

Role dat v konceptu Průmysl 4.0

The Role of Data in the Industry 4.0 Concept

Tomáš Palovský

Bakalářská práce

Vedoucí práce: doc. Ing. Jan Žídek, CSc.

Ostrava, 2021

Poděkování

Rád bych poděkoval panu doc. Ing. Janu Žídkovi, CSc. za pomoc a vedení mé bakalářské práce, dále za poskytování zpětné vazby a cenných rad. Také za veškerou podporu při práci a velmi rychlou komunikaci za všech okolností.

Abstrakt

Průmysl 4.0 je koncept, který udává směr, jímž se ubírají nové technologie v průmyslu. S tímto konceptem je spojena masivní digitalizace výrobního procesu, která otevírá nové možnosti využití digitálních dat pro zefektivnění výroby. Tato práce se v úvodu zabývá právě konceptem Průmyslu 4.0, jeho historickým vývojem a rozbořením klíčových technologií, které jsou s ním spojovány, jako jsou např. "internet věcí" nebo "digitální dvojče". Všechny tyto technologie jsou datově orientované, tj. využívají sběr dat a jejich analýzu k zefektivnění jednotlivých procesů průmyslové výroby. Kvalita a kvantita nasbíraných dat hrají klíčovou roli ve funkčnosti těchto technologií, a právě proto se druhá kapitola zabývá zdroji a typy dat produkovaných ve výrobě. Následná analýza a efektivní využití těchto dat je provedena pomocí softwarových nástrojů, jimž se věnuje třetí kapitola této práce. Mezi tyto nástroje patří např. databázové systémy nebo technologie strojového učení prostřednictvím vyšších programovacích jazyků, jako je např. Python. Cílem této práce je aplikovat nástroje popsané ve třetí kapitole na simulované datové sadě leteckých motorů, zveřejněným NASA, kde pomocí metod strojového učení je vytvořen predikční model, jenž bude předpovídat zbývající životnost motorů na základě dat získaných ze senzorů umístěných v motoru.

Klíčová slova

Průmysl 4.0; typy dat; sběr dat; senzory; python; Jupyter Notebook; prediktivní údržba; Simulovaná datová sada leteckého motoru; strojové učení

Abstract

Industry 4.0 is a concept that sets the direction for new technologies in the industry. This concept is associated with the massive digitization of the production process, which opens up new possibilities to streamline production. In the introduction, this work deals with the concept of Industry 4.0, its historical development, and analysis of key technologies that are associated with it, such as the Internet of Things (IoT) or the digital twin. All these technologies are data-oriented, i.e. they use data collection and analysis to streamline individual processes of industrial production. The quality and quantity of data collected play a crucial role in the functionality of these technologies, so the second chapter deals with the sources and types of data produced in production. Subsequent analysis and effective use of this data are performed using software tools, which are addressed in the third chapter of this work. These tools include, for example, database systems or machine learning technologies through higher-level programming languages such as Python. This work aims to apply the tools described in the third chapter on a simulated dataset of aircraft engines, published by NASA, where using machine learning methods to create a prediction model that will predict the remaining engine life based on data obtained from sensors located in the engine.

Key words

Industry 4.0; data types; Data Collection; sensors; python; Jupyter Notebook; predictive maintenance; Turbofan Engine Degradation Simulation Data Set; machine learning

Seznam použitých symbolů a zkratek

Symbol/zkratka	Význam
PLC	Programmable Logic Controller
IoT	Internet of Things
CPS	Cyber-Physical System
PB	PetaByte
TB	TeraByte
EB	ExaByte
RFID	Radio Frequency Identification
IODD	IO Device Description
RTLS	Real-Time Location System
HMI	Human-Machine Interfaces
SCADA	Supervisory Control and Data Acquisition
SQL	Structured Query Language
MES	Manufacturing Execution System
PC	Personal Computer
ERP	Enterprise Resource Planning
OEE	Overall Equipment Effectiveness
CMMS	Computerized Maintenance Management System
AR	Augmented Reality
VR	Virtuální realita
PLM	Product Lifecycle Management
DSP	Digital Signal Processing
A/D	Analog to digital
PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect express
USB	Universal Serial Bus
CVI	Cortical Visual Impairment
CSV	Comma-Separated Values
VBA	Visual Basic for Applications
SŘBD	System řízení báze dat
TDM	Time Division Multiplex
FFT	Fast Fourier Transform
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RUL	Remaining Useful Life
Li-Ion	Lithium-iontový
IMS	Integrated Management System
PdM	Predictive Maintenance
C-MAPSS	Commercial Modular Aero-Propulsion System Simulation
MIME	Multipurpose Internet Mail Extensions
IDE	Integrated Development Environment
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
SVR	Support Vector Regression
SVM	Support Vector Machines
NumPy	Knihovna funkcí jazyka Python pro vícedimenzionální pole
Pandas	Knihovna funkcí jazyka Python pro datové struktury

Scikit-Learn
Keras
Matplotlib
Seaborn

Knihovna funkcí jazyka Python pro strojové učení
Knihovna funkcí jazyka Python pro hluboké učení
Knihovna funkcí jazyka Python pro vizualizaci
Rozšíření pro knihovnu funkcí Matplotlib

Seznam obrázků

Obrázek 1.1 - vývoj průmyslových revolucí [1].....	11
Obrázek 2.1 - Měřicí řetězec při použití analogového měřicího přístroje	19
Obrázek 2.2 - Měřicí řetězec při použití digitálního měřicího přístroje	20
Obrázek 2.3 - Měřicí řetězec při použití virtuálního měřicího přístroje	20
Obrázek 3.1 - Ukázka zdrojového kódu v prostředí Jupyter Notebook	27
Obrázek 4.1 - diagram vývoje průmyslové údržby [24]	29
Obrázek 4.2 - zjednodušený diagram motoru simulovaného v C-MAPSS [32]	33
Obrázek 4.3 - Rozložení a propojení modulů tak, jak jsou simulovány v C-MAPSS [32]	33
Obrázek 4.4 - Ukázka kódu pro načtení knihoven funkcí	36
Obrázek 4.5 - Kód programu pro načtení dat	37
Obrázek 4.6 - Kód programu pro vytvoření cílové proměnné.....	37
Obrázek 4.7 - Výpis dat z trénovací datové sady	37
Obrázek 4.8 - Ukázky dat ze senzorů vůči zbývajícím životnosti v grafech	38
Obrázek 4.9 - kód programové funkce load_data()	39
Obrázek 4.10 - Korelační mapa	40
Obrázek 4.11 - Indikátory stavu	41
Obrázek 4.12 - kód programové funkce verification()	41
Obrázek 4.13 - kód programové funkce plot_model()	42
Obrázek 4.14 - kód programové funkce model()	42
Obrázek 4.15 - Výsledky lineární regrese	44
Obrázek 4.16 - Výsledky polynomické regrese	44
Obrázek 4.17 - Výsledky SVR	45
Obrázek 4.18 - Výsledky rozhodovacího stromu	45
Obrázek 4.19 - Výsledky náhodného lesa	45
Obrázek 4.20 - Výsledky neuronové sítě	46
Obrázek 5.1 - Výpis předpovězených hodnot oproti skutečným v grafech	47
Obrázek 5.2 - Ukázka dat ze senzorů vůči zbývajícím životnosti v grafech	48
Obrázek 5.3 - Výpis rizika selhání oproti zbývajícím životnosti v grafu [44]	48
Obrázek 5.4 - Ukázka výpisu datové sady referenčního řešení [44]	49

Seznam tabulek

Tabulka 2.1 - Tabulka porovnání klasických versus virtuálních měřicích přístrojů [16].....	23
Tabulka 4.1 - porovnání datových sad	32
Tabulka 4.2 - rozbor datových sad	33
Tabulka 4.3 - Popis senzorů v datové sadě [32]	35
Tabulka 5.1 - srovnání jednotlivých modelů	47

Obsah

Úvod	- 10 -
1 Historický a technologický koncept Průmysl 4.0	- 11 -
1.1 Historický vývoj průmyslových revolucí	- 11 -
1.2 Čtvrtá průmyslová revoluce (Průmysl 4.0)	- 12 -
1.2.1 Klíčové technologie	- 13 -
1.2.2 Smart Factory	- 17 -
2 Zdroje a typy dat produkovaných ve výrobním procesu	- 19 -
2.1 Měřicí řetězce pro sběr dat	- 19 -
2.1.1 Měřicí řetězec analogového měřicího přístroje	- 19 -
2.1.2 Měřicí řetězec digitálního měřicího přístroje	- 20 -
2.1.3 Měřicí řetězec virtuálního měřicího přístroje	- 20 -
2.2 Senzory	- 21 -
2.2.1 Rozdělení a klasifikace senzorů	- 21 -
2.3 Měřicí přístroje	- 22 -
2.3.1 Klasické měřicí přístroje	- 22 -
2.3.2 Virtuální měřicí přístroje	- 23 -
2.3.3 Porovnání virtuálních a klasických měřicích přístrojů	- 23 -
3 Nástroje zpracování dat	- 24 -
3.1 Vývoj nástrojů pro práci s daty	- 24 -
3.1.1 Tabulkové procesory	- 24 -
3.1.2 Databáze	- 25 -
3.1.3 National Instruments DIAdem	- 25 -
3.2 Moderní nástroje pro práci s daty	- 26 -
3.2.1 Python	- 26 -
3.2.2 Jupyter Notebook	- 27 -
4 Případová studie	- 28 -
4.1 Prediktivní údržba	- 28 -
4.1.1 Vývoj průmyslové údržby	- 28 -
4.1.2 Využití prediktivní údržby	- 29 -
4.2 Nalezení vhodných datových sad	- 30 -
4.3 Výběr vhodné konkrétní datové sady	- 31 -
4.4 Popis struktury vybrané datové sady a cíle její analýzy	- 32 -
4.5 Výběr nástroje pro jeho analýzu	- 36 -

4.6	Popis jednotlivých kroků analýzy.....	- 36 -
4.6.1	Import knihoven a načtení dat	- 36 -
4.6.2	Předzpracování dat	- 37 -
4.6.3	Identifikování indikátorů stavu.....	- 39 -
4.6.4	Výběr verifikačních parametrů.....	- 41 -
4.6.5	Trénování predikčního modelu.....	- 42 -
5	Zhodnocení.....	- 47 -
5.1	Dosažené výsledky	- 47 -
5.2	Srovnání s referenčním řešením	- 48 -
5.3	Jak dosáhnout lepšího výsledku	- 49 -
	Závěr.....	- 50 -
	Použitá literatura.....	- 51 -
	Seznam příloh.....	lv

Úvod

Kdo má informace, má moc – je velmi staré a stále platné pravidlo, které je dnes mnohem aktuálnější než kdy dřív. Toto pravidlo platí obecně napříč širokým spektrem oborů a průmysl není výjimkou. Všechny průmyslové revoluce se staly významným milníkem v lidské historii, protože převedly značnou část lidského úsilí na stroje. Průmysl 4.0 je snahou o zdokonalení už tak sofistikovaného procesu. S příchodem Průmyslu 4.0 se markantně zvýšil význam dat nejen v samotném výrobním procesu, ale ve všech aspektech, které s výrobou souvisí, počínaje logistikou a konče pracovními podmínkami zaměstnanců.

Proces získávání informací začíná shromažďováním dat a poté jejich následnou analýzou. Kvalita těchto dat, které se musí zpracovat, je naprosto klíčová, protože jak se říká v Data Science komunitě *Garbage in, garbage out* – tedy že nesmyslný vstup musí vyústit v nesmyslný výstup. To je důvod, proč se zaměřit na to, jak, jaká data a z jakých zdrojů data sbírat a shromažďovat.

Jakmile je nashromážděno dostatek validních dat, tak je nezbytné tato data zpracovat pomocí nástrojů k tomu určených. Popis některých z těchto nástrojů bude také předmětem této práce, je jich totiž celá řada a nesprávná volba při výběru nástrojů může mít velmi negativní následky. Na konci zpracování dat bychom měli získat informace, které dokážeme využít např. k zefektivnění výrobního procesu, zvýšení bezpečnosti nebo ke snížení nákladů. Možností, jak data využívat v náš prospěch je široká škála.

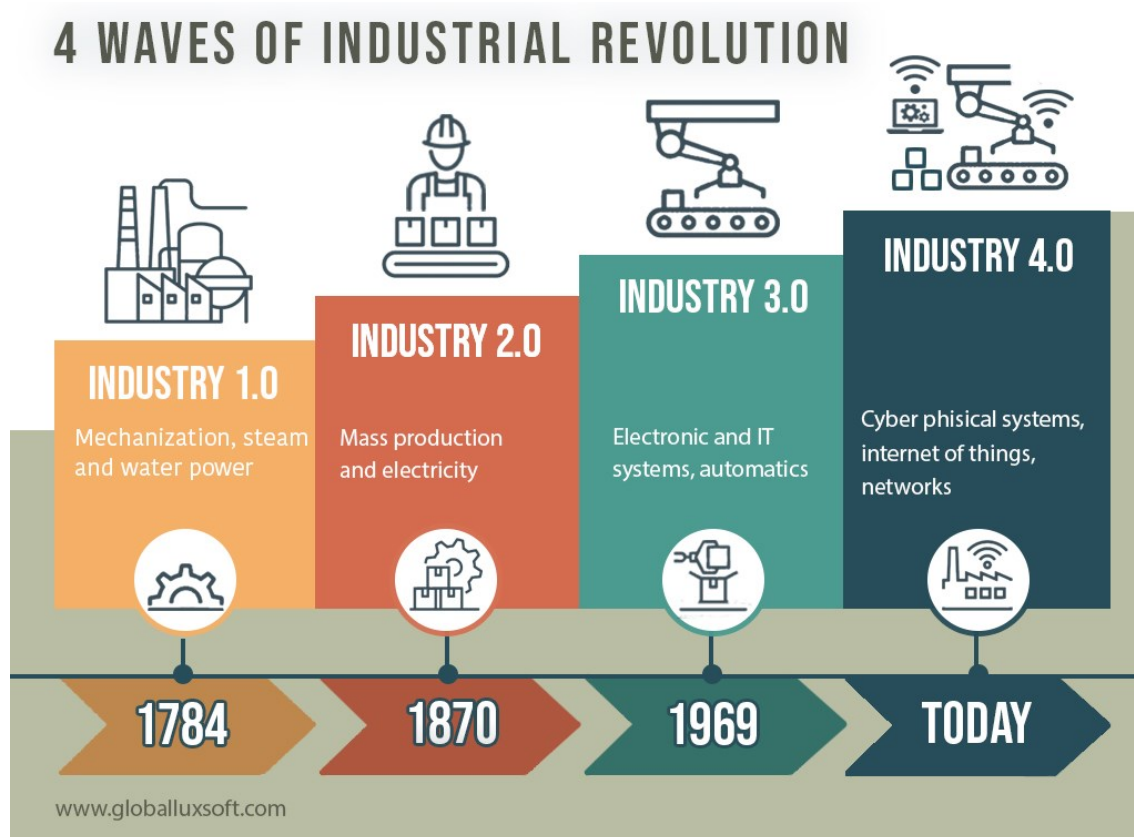
Cílem této práce je využití analýzy dat z výrobního procesu k prediktivní údržbě, což je analytická metoda, která nám umožňuje předpovídat, kdy dojde k selhání výrobních zařízení a předcházet těmto selháním včasnou a efektivní údržbou výrobních zařízení.

Teoretická část bakalářské práce je věnována popisu a rozboru konceptu Průmysl 4.0 se zdůrazněním role dat, kterou v něm hrají. V následující kapitole je popsáno, jak se tato data získávají a jaké typy dat nebo v jaké formě můžeme data shromažďovat. Poslední část teoretické kapitoly bakalářské práce popisuje vybrané nástroje na zpracování dat.

V praktické části bakalářské práce je provedena analýza dat za účelem prediktivní údržby v rámci vybrané případové studie. Praktická část bakalářské práce je rozdělena do tří částí. První část se zabývá výběrem dat, která budou zpracována, následně druhá část řeší samotnou analýzu dat a poslední část je věnována vizualizaci a vyhodnocení výsledků analýzy dat do podoby validní a využitelné informace.

1 Historický a technologický koncept Průmysl 4.0

1.1 Historický vývoj průmyslových revolucí



Obrázek 1.1: vývoj průmyslových revolucí [1]

1. Průmyslová revoluce

První průmyslovou revolucí nazýváme období mezi léty 1770 a 1830, kdy proběhl v Anglii jako v první zemi na světě hospodářský rozmach díky objevení parní energie a mechanizaci výroby, během kterého došlo jak ke změně krajiny a země, tak ke změně ve všech oblastech života obyvatel. Typickými stavbami byly továrny s vysokými kouřícími komíny, v jejichž okolí vyrostla sídla pro masu továrních zaměstnanců.

Symbolem průmyslové revoluce se stal především parní stroj, který zdokonalil v 60. letech 18. století James Watt. Až po tomto zdokonalení mohlo dojít k jeho masovému nasazení v různých oblastech od dolů a hutí až po parní dopravu (lokomotivy).

Mezi důsledky průmyslové revoluce patří rozvoj výroby, a to napříč spektrem oborů, jako například rozvoj strojírenství, chemie, dopravy a obchodu. Dalším důsledkem je celková změna společenské struktury, růst počtu obyvatel, snížení úmrtnosti a urbanizace.[2]

2. Průmyslová revoluce

Slovo „revoluce“ je dobré, poněvadž podněcuje zvědavost, vzbuzuje emoce a šokuje. Proto i další výraznější změny, ke kterým došlo v průmyslu, začaly být zpětně nazývány „revolucemi“. To je případ i tzv. 2. průmyslové revoluce, která začala v 19. století objevem elektrické energie a zavedením montážních výrobních linek.[3]

Henry Ford (1863–1947) si nápad hromadné výroby přinesl z jatek v Chicagu: Prasata byla zavěšená na pásové dopravníky a každý řezník vykonával pouze část bourání zvířete. Henry Ford tento princip převedl do prostředí automobilové výroby a drasticky změnil její proces. Zatímco dříve jedna stanice kompletovala celý automobil, nyní byly vozy vyráběny po jednotlivých krocích na lince – což bylo výrazně rychlejší a navíc levnější.[4]

3. Průmyslová revoluce

Třetí průmyslová revoluce začala v 70. letech 20. století, ta bývá nejčastěji spojována s automatizací, elektronikou a rozmachem informačních technologií. Stejně jako byl přechod od uhlí a páry k elektřině poměrně spojitý a logický, tak i přechod od mechanismů k automatům byl spíše výsledkem přirozené evoluce než skutečnou revolucí. Za její počátek se nejčastěji uvádí rok 1969, kdy byl vyroben první programovatelný logický automat čili PLC. Jedná se vlastně o malý průmyslový počítač, řídicí jednotku, pro automatizaci procesů v reálném čase.[3]

S nástupem 3. Průmyslové revoluce je také spojen pojem data, která slouží jako prostředek průmyslové automatizace výrobního procesu, ale role dat a jejich zpracovávání značně vzrostl až s příchodem konceptu Průmyslu 4.0. Nyní se nacházíme na přelomu další průmyslové revoluce, kdy trendy automatizace jsou naprosto běžné a podnikatelé hledají možnosti, jak snižovat náklady a zvyšovat zisky. Cesta je Průmysl 4.0, který představuje zařazení několika datově založených technologií, které zdokonalují celý výrobní proces.

1.2 Čtvrtá průmyslová revoluce (Průmysl 4.0)

Pojem Industry 4.0 byl oficiálně definován v dokumentu, který byl představen na veletrhu v Hannoveru v roce 2011. Tento nový průmyslový model zahrnuje velké změny v současných průmyslových paradigmatech a snaží se o bezešvé propojení mezi reálným a digitálním světem. To znamená využívání mnoha nových nástrojů a technologií jako je např. široké využití internetu věcí (IoT), zakomponování kyber-fyzických systémů nebo použití senzorů pro vyhodnocování celého procesu v reálném čase.[5]

Koncept průmysl 4.0 představuje zařazení několika vlastností do výrobního procesu, jejichž společným jmenovatelem je využívání dat ve velkém měřítku. Mezi tyto vlastnosti patří:

- **Interoperabilita** - Je to schopnost kyber-fyzických systémů (CPS) vzájemně spolupracovat, poskytovat si informace a dosáhnout součinnosti prostřednictvím internetu věcí (IoT)
- **Virtualizace** - Virtualizace vytváří virtuální simulaci výpočetního prostředí namísto toho fyzického. Virtuální kopie smart factory (Digital twin) propojuje data ze senzorů spolu s virtuálními a simulačními modely továrny. [6]
- **Decentralizace** - Na rozdíl od centralizovaných řídicích systémů, kdy řídicí aktivitu zajišťuje jeden výkonný řídicí prvek, se v decentralizovaném kyber-fyzickém systému schopnost

jednat deleguje na jednotlivé autonomní prvky v rámci smart factory, které jsou schopné vyhodnocovat situace v reálném čase na základě získaných dat a zasahovat do výrobního procesu v reálném čase.

- **Orientace na služby** - produkt si říká, jaké služby na něm mají být ve výrobních buňkách provedeny
- **Modularita** - schopnost adaptace smart factory na změnu požadavků výměnou nebo rozšířením individuálních modulů. Výrobní linka se přizpůsobuje individuálním požadavkům zákazníka v rámci hromadné výroby – to vyžaduje implementaci následujících principů do konstrukce výrobní linky: Schopnost vlastní automatické optimalizace (self-optimization), Schopnost vlastní automatické konfigurace (self-configuration), Schopnost vlastní automatické diagnostiky stavu (self-diagnostics), Schopnost učení se s využitím umělé inteligence (cognition), Schopnost poskytovat obsluhu inteligentní podporu (intelligent support of workers) - mnohé z těchto schopností se opírají o digitalizaci. Zpracování výsledků měření, simulace – vyvstává rostoucí potřeba analýzy dat.

1.2.1 Klíčové technologie

Velká data a pokročilá analýza

Ve výrobním procesu procesu vzniká obrovské množství dat použitelných pro jejich analýzu s cílem zefektivnění tohoto procesu. Množství dat, které se v celém průmyslu ročně vyprodukuje, každým rokem exponenciálně roste. Pojem Velká data definuje takové množství dat, které už není možné zpracovávat klasickými prostředky za "rozumný" časový úsek. Rozumným časovým úsekem se myslí doba v řádech hodin, dnů nebo týdnů. Pro porovnání, kdybychom nechali zpracovávat velká data na běžném počítači, trvala by taková analýza několik let až desítek let, tak obrovská Velká data jsou.[7]

Velká data jsou charakterizována pomocí čtyř vlastností, která začínají na písmeno V:

- **Volume (objem)** - Společnosti shromažďují data z nejrůznějších zdrojů, jako jsou mimo jiné účetnictví, chytrá zařízení, průmyslové vybavení, sociální média, bezpečnostní systémy a zálohování. Big data mohou dosahovat objemů řádově v petabytech (1 PB = 1000 TB) až exabytech (1 EB = 1000 PB).[8]
- **Velocity (rychlost)** - V takovém objemu dat je rychlost jeho přenášení klíčová. RFID tagy, senzory a inteligentní měřicí přístroje vedou k potřebě řešit tyto přívaly dat v téměř reálném čase a s příchodem 5G sítí budou tyto datové toky proudit nebývalou rychlostí. [9]
- **Variety (různorodost)** - Data mohou přicházet v jakémkoliv formátu, od strukturovaných numerických dat v tradiční databázi, přes nestrukturované textové dokumenty, emaily až po audio či videonahrávky. [9]
- **Veracity (věrohodnost)** - Je nutné při analýze zohlednit kvalitu dat z různých zdrojů, protože může dojít k chybě v systému nebo šumu signálu na nějakém senzoru. Je obtížné tyto data čistit, porovnávat a provázivat napříč systémem. [9]
- **Variability (proměnlivost)** - Společnost SAS, která se zabývá problematikou Big data, přidala 5. vlastnost do této charakteristiky a tou je proměnlivost. Kromě rostoucích rychlostí a rozmanitosti dat jsou datové toky nepředvídatelné – často se mění a velmi liší. Je to náročné, ale firmy potřebují vědět, kdy a co je na trhu trendy a jak zvládat denní, sezónní a událostmi vyvolané špičky dat. [9]

Senzory nové generace

Senzorová technika obecně patří k hlavním pilířům, na kterých je dnešní průmysl postavený. Senzory připojené k zařízením nám pomáhají monitorovat a shromažďovat odlišné datové body v konkrétních fázích výrobního procesu, dále nám poskytují přehled o výrobku během výroby v reálném čase a mnohem více.

Velkou hnací silou Průmyslu 4.0 je trend zlevňování senzorové technologie, což umožňuje měřit obrovské množství parametrů nejen v procesech výroby, ale ve všem, co nám poskytne lepší monitorování celé továrny jako celku.

Moderní senzory využívají technologii IO-link, což je speciální digitální komunikace pro snadné připojení a komunikaci různých snímačů a případně akčních členů po sériové sběrnici. Mezi hlavní výhody této technologie patří: [10]

- **Jednoduchá kabeláž** - Spojení mezi hlavním zařízením a zařízením IO-link je povoleno až do maximální délky 20 m s nestíněným 3vodičovým kabelem. [5]
- **Snadné ovládání a automatické nastavení parametrů** - Každé zařízení IO-link má přidružený IODD (IO Device Description). Tento soubor obsahuje informace o výrobci, čísle článku, funkčnosti, atd. Tyto informace může uživatel snadno přečíst a zpracovat. [5]
- **Neomezená rozšíření** - S jedním portem IO-link lze propojit až 16 standardních bistabilních senzorů / akčních členů pomocí tří žilového kabelu o délce až 20 metrů. Tímto způsobem jsou v době instalace drasticky redukovány prvky, jako jsou vodiče, svorky, rozhraní, spojovací skřínky, kabelové řetězy atd. [5]
- **100% kompatibilní** - Konsorcium IO-link pokrývá největší světové dodavatele technologie průmyslové automatizace. Již existují zařízení IO-link pro připojení senzorů a akčních členů k nejoblíbenějším sítím Fieldbus Profibus, Profinet, Ethernet I / P, Ether-cat, CC-link atd. [5]

IoT (Internet of things)

IoT neboli internet věcí je koncept založený na propojení jakýkoliv zařízení mezi sebou. Cílem je umožnit těmto zařízením posílat data přes připojení k síti za využití IPv6 protokolu bez zásahu člověka a tím je udělat více inteligentní a nezávislé.

Revoluce v oblasti IoT se očekává s příchodem 5G sítí, které díky své vysoké rychlosti a kapacitě umožní zvýšit počet připojených zařízení k internetu v dosud nevídaném měřítku. Typicky půjde o rozsah 24–86 GHz, ale pro Evropu je schválené pásmo 24,25–27,5 GHz. Tato spektra jsou prakticky nevyužitá a umožňují nasadit 400 MHz široké kanály. To dovoluje dosahovat na přenosové rychlosti dosud netušených hodnot – až 20 Gb/s s odezvou 1 ms. [11]

RFID Tagy (Radio Frequency Identification) a RTLS (Real Time Location System)

RFID tagy jsou malá zařízení podobná štítku, která mohou být připojena nebo zabudována do jakéhokoli objektu. Obsahují antény pro přijímání a odesílání žádostí rádiové frekvence na RFID vysílači.

Jejich výhodou je, že nemusí mít přímý vizuální kontakt mezi vysílačem a přijímačem a na rozdíl a dokážou uchovat více informací než např. čárové kódy. RFID je jednou z technologií která se využívá v RTLS systémech.

RTLS se používají k automatické identifikaci a sledování polohy objektů nebo osob v reálném čase, obvykle v budově nebo ohraničené oblasti. Pevné referenční body přijímají bezdrátové signály ze značek RFID k určení jejich polohy.[5]

Autonomní vozidla

Jsou to vozidla, která nepotřebují k jízdě řidiče. Nejen v průmyslu jsou využívána běžně pro přemísťování materiálů mezi různými lokacemi. Nebo mohou být využívána v prostředí, které není pro člověka bezpečné.

Je několik způsobů jak mohou být tyto vozidla bezpečně naváděna po prostoru. Např. pomocí umělého vidění navádí systém vozidlo po trase, která je namalovaná na zemi.[5]

HMI (Human-Machine Interfaces)

HMI je zařízení nebo systém. Který funguje jako rozhraní mezi člověkem a strojem. V posledních letech se k tomuto nejvíce používají dotykové obrazovky, přes které může uživatel ovládat daný stroj, něco celý systém. V současnosti dochází v pokroku ve výrobě bezkontaktních zařízení (Touchless). [5]

SCADA (Supervisory Control and Data Acquisition)

Je to systém, jehož účelem je dohlížet na výrobní procesy na velkou vzdálenost. Slova SCADA a HMI často vyvolávají zmatek. Rozdíl spočívá v monitorovací funkci, kterou mohou systémy SCADA provádět, aby na monitoru bylo možné sledovat vývoj proměnných, které kontrolujeme. [5]

SCADA systémy operují na úrovni nad hardwarem (PLC, senzory, ...). Dále umožňují sběr a následné ukládání dat mnoha způsoby od textových souborů až po SQL databáze, které ukládají data s obrovskou rychlostí. Trendem je možnost vzdáleného přístupu mimo klasický PC, tj. monitorování a řízení provozu z tabletu nebo smartphonu. [12]

MES (Manufacturing Execution System)

Systém MES je softwarový nástroj, který funguje jako rozšíření systému ERP (Enterprise Resource Planning), ale je orientován na plánování a provádění výroby. Zatímco systém ERP určuje, co se má vyrábět, poskytuje systém MES nezbytné funkce pro správu klíčových oblastí závodu, jako jsou lidé, materiály, procesy, kvalita, sledovatelnost, údržba a vizuální továrna. Bez papírů, s přesností a v reálném čase. MES umožňuje automatické snímání výrobních dat, monitorování operací a identifikaci odpadu a oblasti zlepšování výkonu (OEE) a doby realizace.[5]

CMMS (Computerized Maintenance Management System)

Jedná se o softwarový nástroj, který pomáhá při řízení služeb údržby společnosti. Cílem je řídit pomocí CMMS procesy nápravné, preventivní a prediktivní údržby ve výrobním zařízení, ale také kontrolovat zásoby, nákupy a smlouvy. S tímto modelem je možné mít globální vizi výkonu údržby, měřit efektivitu oddělení údržby v každé linii, s náklady, dobami zastavení a dobami zásahu. [5]

Aditivní výroba

3D tisk je aditivní proces, který umožňuje vytvářet objekty vrstvu po vrstvě zdola nahoru. Přesná technologie závisí na materiálech, estetice, mechanických vlastnostech a výkonu, který potřebujete. Existuje možnost výroby od nízkonákladového prototypu po finální model s vysokým

rozlišením a kvalitou s vlastnostmi podobnými finálnímu produktu. Nejvýraznějším rysem této technologie, na rozdíl například od obráběcích procesů, je to, že čím složitější je design, tím je výroba levnější. [5]

Rozšířená realita

Skládá se ze sady zařízení, která přidávají virtuální informace k existujícím fyzickým informacím, tj. Přidávají virtuální syntetickou část ke skutečnému. Rozšířená realita (RA) se liší od virtuální reality (VR), ve které se uživatel izoluje od materiální reality fyzického světa, aby se ponořil do zcela virtuálního prostředí. Prostřednictvím rozšířené reality začleňujeme data a digitální informace do reálného prostředí prostřednictvím rozpoznávání vzorů vytvářených prostřednictvím konkrétního softwaru. [5]

Kolaborativní roboti

Kolaborativní roboti se vyznačují tím, že jsou lehcí, flexibilní a snadno se instalují. Jsou navrženi pro interakci s lidmi ve sdíleném pracovním prostoru bez nutnosti instalace bezpečnostních plotů. Kolaborativní roboti sami nezaručují, že konečná aplikace je kolaborativní. To závisí na mnoha dalších faktorech souvisejících s procesem.

Jejich malá velikost, univerzálnost a dostupná cena je odlišuje od tradičních průmyslových robotů a činí je vhodnými nejen pro velkou společnost, ale také pro malé a střední podniky, které potřebují nákladově efektivní a snadno použitelná řešení. [5]

Umělá inteligence

Nejdůležitější snahou umělé inteligence je napodobit lidské rozhodování ale mnohem rychleji. Abychom mohli mít umělou inteligenci, potřebujeme Velká data, Internet věcí a algoritmy které stroji umožní rozpoznat vzorce chování a na základě těchto vzorců schopnost se rozhodovat zcela automaticky. [5]

Cloud Computing / Hosting

Cloud computing je služba, která umožňuje přesun výpočetního procesu mimo výrobní proces. Tedy společnost nemusí utrácet velké množství peněz za výpočetní servery, namísto toho stačí rychlé připojení k internetu. [5]

Kybernetická bezpečnost

Celý koncept Průmyslu 4.0 je datově založený, což znamená, že všechny systémy, které ve společnosti fungují, jsou nějakým způsobem propojeny a operují s obrovským množstvím dat. Úspěšný kybernetický útok by mohl mít za následek od odcizení citlivých dat zaměstnanců, až po úplnou paralýzu celé továrny. Proto je důležité dbát na zabezpečení před útoky na všech úrovních a je nezbytné mít v organizaci strategii: [5]

- Vypracování pravidel a postupů pro každou službu organizace
- Definování akcí, které mají být podniknuty, a výběr osob, které mají kontaktovat v případě zjištění možného vniknutí.
- Citlivost operátorů na problémy spojené s bezpečností počítačových systémů.

Kolaborativní platformy

Kolaborativní platforma je virtuální pracovní prostor, který centralizuje funkce řízení a znalosti projektu a je přístupný všem zúčastněným zaměstnancům. Platforma tedy umožňuje realizaci konkrétní práce, která lze sdílet s ostatními uživateli pomocí různých nástrojů integrovaných do samotné platformy: systém výměny souborů, fórum, chat atd.

Mezi nejpoužívanější platformy patří Yammer od společnosti Microsoft a Trello. Dále tu jsou Producteev, SocialShared a Active Collab. [5]

PLM (Product Lifecycle Management)

PLM je nové podnikové počítačové řešení, jehož cílem je se zaměřit a kontrolovat celý životní cyklus produktu, od jeho návrhu, až po jeho prodej. A když je produkt inteligentní a propojený, data generovaná během celého životního cyklu již nekončí dodáním produktu. Nové technologie internetu věcí umožňují společně získávat a používat informace o výkonu produktu během jeho provozu s potenciálem vylepšovat budoucí produkty. [5]

Digitální dvojče

Digitální dvojče je softwarový model, který má digitálně duplikovat fyzickou podobu výrobního procesu. Využívají data ze senzorů nainstalovaných na fyzických strojích reprezentující jejich status, pracovní podmínky nebo pozici v reálném čase.

Největší výhodou tohoto modelu je simulace a testování změn ve výrobním procesu, aniž by byl, jakkoliv narušen. Pokud bude všechno fungovat ve virtuálním prostředí, tak se můžou aplikovat změny na reálný výrobní proces tzv. virtuální uvádění do provozu - virtual commissioning. [5]

Kyber-fyzické systémy (CPS)

Jsou to systémy, ve kterých počítače interagují přímo s fyzickým světem. Příkladem CPS jsou automobily, ve kterých řídicí jednotky ovládají nejen motor, brzdění, kontrolu trakce nebo stabilitu vozidla, ale také napomáhají řidiči v jízdě různými podpůrnými systémy např. parkovací asistent.

CPS jsou založeny na propojení několika konceptů, nástrojů a systémů popsaných výše. Tyto systémy integrují 3 základní pilíře Průmyslu 4.0: [5]

- 1) Horizontální integrace informačních systémů, které se používají v různých článcích dodavatelského řetězce.
- 2) Digitální integrace inženýrských procesů (Product lifecycle management)
- 3) Vertikální integrace informačních systémů (senzory, řídicí systémy, SCADA systémy, digitální dvojče, ...)

1.2.2 Smart Factory

Je to koncept pro vyjádření uplatnění digitalizace ve výrobě, také to je bezpečný a čistý prostor, ve kterém se nepřetržitě shromažďují a sdílí data prostřednictvím připojených strojů, zařízení a výrobních systémů. Data pak mohou být použita pro optimalizaci a zlepšení výrobního procesu nebo k reakci na nové požadavky. Díky klíčovým technologiím Průmyslu 4.0, které mohou být naimplementovány dle potřeby, jsou inteligentní výrobní postupy plně komplexní. To všechno má

příspěvek ke snížení výrobní ceny a také k větší flexibilitě, kvalitě, rychlosti a komunikaci na všech úrovních. [5]

Požadavky na Smart Factory

Automatizace - pro zvýšení rychlosti v opakujících se procesech, za pomoci kolaborativních robotů, kteří provádějí úkoly bez přidané hodnoty a snadno se přizpůsobí jinému úkolu.

Digitální propojení - zahrnuje masivní použití senzorické techniky pro vyhodnocování co největšího množství dat v reálném čase pomocí umělé inteligence a zefektivnit celý proces výroby

Flexibilita - možnost upravovat proces pro konkrétní požadavky zákazníka.

Udržitelnost - s racionálním a odpovědným využíváním zdrojů a energie.

Lidskost - jedna z nejdůležitějších částí je dbát na dobrou kvalifikaci zaměstnanců a podporovat jejich rozvoj. Dále je také důležité, aby firma vytvářela příjemné prostředí pro své zaměstnance.[5]

Čtyři úrovně Smart Factory

- **Úroveň první: dostupná data**

To je pravděpodobně současný stav většiny továren. Data jsou k dispozici, ale nejsou přístupná. Třídění a analýza dat vyžadují manuální práci, a to může být časově velmi náročné, což má za následek neefektivnost procesu zlepšování výroby, než jak bylo zamýšleno nebo potřeba. [13]

- **Úroveň druhá: Přístupná data**

V této fázi jsou data prezentována v lépe stravitelné formě. Data jsou strukturálně uspořádána a správně tříděna na jednom místě s dalšími systémy, které pomáhají vizualizovat data a zobrazovat řídicí panely. Továrna je schopna provádět proaktivní analýzu, i když to může ještě vyžadovat určitý čas a úsilí. [13]

- **Úroveň třetí: Aktivní data**

Aktivní data znamenají data, která mohou provádět proaktivní analýzu pomocí strojového učení a umělé inteligence k vytváření přehledů bez velkého lidského dohledu. Systém dokáže připnout klíčové problémy a anomálie k předpovědi selhání s vysokou přesností a informovat relevantní lidi s cennými poznatky ve správný čas. [13]

- **Úroveň čtvrtá: Akčně orientovaná data**

V této fázi může strojové učení generovat použitelná řešení problémů, které jsou identifikovány v dřívějších fázích. Výrobní stroje a zařízení, které jsou připojeny k tomuto modulu nebo systému, mohou tyto změny provést bez lidského zásahu. Shromažďování dat, identifikace problémů a generování řešení probíhá postupně s malým nebo žádným vstupem člověka. [13]

2 Zdroje a typy dat produkovaných ve výrobním procesu

V průmyslu je mnoho technologií, ve kterých mají data sbírána ve výrobním procesu velmi důležitou roli, jako jsou mimo jiné technologie:

- Řízení výrobních zařízení
- Monitoringu stavu výrobních zařízení
- Testování produktů
- Digitálního dvojčete

Všechny tyto technologie využívají sběr dat a jejich analýzu, aby to bylo možné s využitím výsledků analýzy těchto dat optimalizovat infrastrukturu výrobního procesu. Mezi prvky potřebné pro optimalizaci této infrastruktury také patří:

- Smart senzory
- Internet of Things
- Cloud computing
- Umělá inteligence

[14][15]

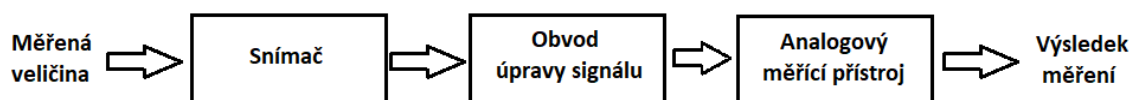
2.1 Měřicí řetězce pro sběr dat

Měření lze definovat jako soubor činností, jejichž cílem je stanovit hodnotu veličin. Postup pro dosažení tohoto cíle je měřicí metoda. Tyto měřicí metody můžeme dělit na metody přímé a nepřímé. Přímá měřicí metoda vychází z definice měřené veličiny (měřicí přístroj indikuje přímo měřenou veličinu). Nepřímá měřicí metoda vychází z určení funkční závislosti měřené veličiny na jiné měřené fyzikální veličině a k výsledku měření se dochází aplikací této funkční závislosti, přičemž nezávisle proměnnými v této funkční závislosti jsou konstanty a přímo měřené fyzikální veličiny.

Nástrojem k naplnění cíle měření je měřicí řetězec, který realizuje zjištění hodnoty měřené veličiny. Vstupní veličinou do měřicího řetězce je měřená veličina, výstupní veličinou je výsledek měření. Jednotlivé články měřicího řetězce získávají, upravují, přenášejí či zpracovávají informace o měřené veličině. [14][15]

2.1.1 Měřicí řetězec analogového měřicího přístroje

Obecné schéma měřicího řetězce se mění podle typu použitého měřicího systému. Historicky nejstarší je měřicí řetězec používající analogového měřicího přístroje, jehož strukturu ukazuje následující obrázek:



Obrázek 2.1: Měřicí řetězec při použití analogového měřicího přístroje

Na vstupu tohoto řetězce je měřená veličina a následně snímač, který má za úkol převést tuto veličinu na signál (nejčastěji napěťový), který bude po úpravě zpracován analogovým přístrojem. Úkolem obvodu úpravy signálu je upravit výstupní signál snímače tak, aby mohl být bezpečně zpracován měřicím přístrojem. Mezi jeho hlavní patří:

- Galvanické oddělení měřicího přístroje od místa měření
- Úprava velikosti signálu
- Odstranění šumu v signálu

Analogový měřicí přístroj může pak měřit tuto fyzikální veličinu přímo, nebo nepřímo pomocí matematických operací.

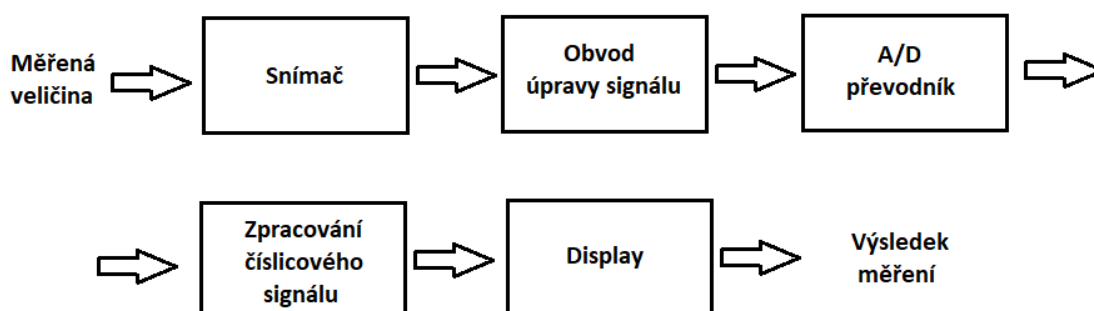
[14][15]

2.1.2 Měřicí řetězec digitálního měřicího přístroje

Modernější měřicí metodou, je použití digitálního měřicího přístroje. Digitální měřicí přístroje přináší v praxi výhody, mezi které také patří:

- lepší odolnost proti superpozici rušivých složek signálů
- možnost snadného zpracování signálů v jejich digitální podobě

Při použití digitálního měřicího přístroje se blokové schéma měřicího řetězce mění do následující obecné podoby:



Obrázek 2.2: Měřicí řetězec při použití digitálního měřicího přístroje

Začátek schématu je stejný jako u analogového přístroje, ale po úpravě signálu následuje jeho převod na číslicový signál A/D převodníkem. Číselná reprezentace signálu umožňuje jeho následné zpracování (DSP – Digital Signal Processing) za použití mikroprocesorů, jednočipových mikropočítačů či signálových procesorů. Výsledek zpracování číslicového signálu je potom zobrazen na displeji (v číslicovém nebo grafickém tvaru) jako výsledek měření. [14][15]

2.1.3 Měřicí řetězec virtuálního měřicího přístroje

Nejmodernější metodou měření je použití virtuálního měřicího přístroje.



Obrázek 2.3: Měřicí řetězec při použití virtuálního měřicího přístroje

Schéma pro virtuální měřicí přístroj se podobá schématu pro digitální měřicí přístroj, avšak u virtuálního měřicího přístroje nahrazuje bloky zpracování signálu a displej personální počítač. Největší výhodou u virtuálních měřících přístrojů je software, jenž definuje funkčnost měřicího systému. [14][15]

2.2 Senzory

Senzory jsou nejběžnějším nástrojem pro získávání dat z reálného prostředí do prostředí virtuálního. Vzhledem k velikému rozpětí technických aplikací vyžadujících měření fyzikálních veličin existuje také obrovské množství různých druhů senzorů. [14][15]

2.2.1 Rozdělení a klasifikace senzorů

Snímače neelektrických veličin lze klasifikovat podle několika kritérií. Nejviditelnější je kritérium měřené veličiny. Podle druhu vstupního signálu se rozeznávají senzory:

- Mechanické (délka, poloha, rychlost, zrychlení, úhlové veličiny – natočení, rychlost a zrychlení, síla, tlak, mechanická deformace, ...)
- Tepelné (teplota, tepelný tok, ...)
- Elektrické (napětí, proud i jejich frekvence a fáze, odpor, kapacita, indukčnost, permitivita)
- Magnetické (indukce, intenzita a jejich tok, ...)
- Optické (intenzita, energie, barva záření, ...)
- Akustické (intenzita, frekvence, akustický tlak, ...)
- Chemické (koncentrace, aktivita, pH, výhřevnost, ...)
- Biologické (bioproudy a biopotenciály, aktivita, ...)
- Radiační (intenzita záření, počet částic a druh záření, ...)

Senzory (a převodníky obecně) transformují jeden určitý druh signálu na jiný, nejčastěji elektrický signál. Podle toho, zda se při převodu přímo využívá část energie měřené veličiny, nebo zda je nutný další (pomocný) zdroj, se senzory označují jako:

- Aktivní neboli generátorové, které „generují“ elektrickou energii nebo náboj (termoelektrické, piezoelektrické)
- Pasivní neboli modulační, kde měřená veličina moduluje signál z pomocného zdroje (např. změnou odporu, kapacity apod.)

Hledisko výstupního signálu je velmi důležité vzhledem k dalšímu zpracování a využití výsledku měření. Podle druhu výstupního signálu se senzory dělí na:

- Analogové
- Digitální

Výstup digitálního senzoru má formu digitální - posloupnost logických stavů nebo dat v daném přenosovém protokolu. Digitální senzory obsahují A/D převodník a jejich výstupní digitální signál se přenáší snadněji než signál analogový. Digitální výstup má také lepší opakovatelnost a odolnost proti rušení a bývá mnohdy i přesnější. Avšak existuje pouze málo fyzikálních veličin, které mohou být přímo měřeny digitálními senzory. V současnosti naprostá většina senzorů, označovaných jako digitální, využívá dodatečné digitalizace původně analogového elektrického signálu. [14][15]

Další kritériem klasifikace senzorů je typ proměnné veličiny v senzoru, která se mění působením vstupního signálu.

- Odporové
- Kapacitní
- Indukční
- Napěťové
- Proudové
- Nábojové

Z pohledu historického vývoje měřicí techniky se v odborné literatuře rozlišuje několik generací senzorů:

- první generace představuje klasické senzory využívající základní měřicí principy (odporové, kapacitní, indukční, ...)
- druhá generace je charakteristická využitím polovodičových technologií, které umožňují kombinaci snímacích prvků a elektroniky, kombinaci více typů senzorů do jednoho celku a celkové zlepšení parametrů
- třetí generace využívá optické a optoelektronické principy, optické vláknové technologie
- nastupující čtvrtá generace je založena na spojení biologických elementů a elektronických obvodů a umožňuje rozeznávání a sledování koncentrací specifických molekul.

Další vývoj senzorové techniky směřuje k nano systémům, využívajícím extrémní miniaturizaci, inteligentní materiály a kvantově-mechanické jevy. [14][15]

2.3 Měřicí přístroje

Vývoj měřicí techniky

Výrobci měřicí techniky se v tržním hospodářství řídí stejnými pravidly jako kdokoli jiný na trhu. Jejich snahou je maximalizovat svůj zisk ovládnutím co největší části trhu svými výrobky. Pro dosažení tohoto cíle musí získat co nejvíce zákazníků se zájmem o své výrobky. Tento zájem bude narůstat tím strměji, čím lepší bude uspokojení potřeb těchto zákazníků právě prostřednictvím produkce těchto výrobců.

Definujme pojem flexibilita měřicí techniky obecně jako schopnost uspokojení potřeb koncového uživatele, která je přímo úměrná zastoupení SW ve struktuře měřicí techniky. V celosvětovém měřítku sleduje vývojový trend měřicí techniky zvyšující se flexibilitu měřicích systémů. [16]

2.3.1 Klasické měřicí přístroje

Klasické měřicí přístroje, ať už analogové nebo digitální, které mají samy o sobě schopnost provádět měření, jsou nazývány samostatně stojící přístroje (stand alone instruments). Takto označované přístroje jsou vždy vybaveny ovládacími prvky (tlačítka, přepínači...) a prvky indikačními (ručkové ukazatele, displeje, obrazovky...) tak, aby mohly být člověkem ovládány, a měřené údaje z nich člověkem čteny. Samostatně stojící přístroje mohou, ale také nemusí být vybaveny komunikačním rozhraním, pro připojení k počítači. [16]

2.3.2 Virtuální měřicí přístroje

Podstatou virtuálního přístroje je doplnění otevřené architektury počítače tím, co mu chybí, aby mohl plnit úlohu měřicího přístroje. V oblasti hardwaru je to zásuvná multifunkční karta vybavená buď konektorem pro zasunutí této karty do systémové desky počítače (PCI, PCIe sběrnice), nebo kabelem pro připojení k některému ze standardních rozhraní počítače (USB, FireWire, Ethernet). V oblasti softwaru je to vhodný program pro počítač, který realizuje všechny funkce měřicího přístroje a plní tak úlohu firmwaru měřicího přístroje. Zabraňuje se tak i duplicitě, které se objevuje při propojení měřicího přístroje a počítače. [16]

Přístrojové ovladače

Přístrojové ovladače slouží jako knihovna funkcí, které podporují daný měřicí přístroj. Koncový uživatel má díky tomu snadný přístup ke všem funkcím, které jsou podporovány přes komunikační rozhraní.

Přístrojové ovladače jsou dnes k dispozici ke stovkám měřicích přístrojů světových výrobců, a to jak pro textově orientované vývojové prostředí CVI, tak pro graficky orientované vývojové prostředí LabVIEW.

2.3.3 Porovnání virtuálních a klasických měřicích přístrojů

Filozofie virtuálních měřicích přístrojů je velmi progresivní. Umožňuje při zachování výkonnostních parametrů klasické měřicí techniky vytvářet přístroje, jejichž požadavky přesně odpovídají požadavkům uživatele, protože funkce těchto měřicích přístrojů jsou realizovány softwarově. Navíc tato koncepce umožňuje doplňovat funkčnost přístroje o další funkce podle narůstajících a měnících se potřeb koncového uživatele. Cena takto vytvářených měřicích přístrojů bývá srovnatelná nebo nižší, než je cena klasické měřicí techniky. [16]

Tabulka 2.1: Tabulka porovnání klasických versus virtuálních měřicích přístrojů [16]

Hledisko	Tradiční přístroj	Virtuální přístroj
Funkce definuje	Výrobce	Uživatel
Orientace přístrojů a jejich propojitelnost	Specifické podle funkcí, hlavně používané samostatně s omezenou propojitelností	Specifické podle aplikace, propojitelnost s návazností na počítačové sítě a periférie
Klíčový komponent	Hardware	Software
Náklady na výrobu a opakovatelnost použití	Vysoké	Nízké, opakovatelně použitelné
Architektura	Uzavřená, pevná množina funkcí	Otevřená, pružná funkčnost odrážející výpočetní techniky
Návaznost na technologický proces	Pomalá (doba obměny 5-10 let)	Rychlá (doba obměny 1-2 roky)
Náklady na vývoj a údržbu	Vysoké, horší poměr cena/výkon	Nízké, dobrý poměr cena/výkon

3 Nástroje zpracování dat

Zpracování dat (data processing) je souhrnné označení pro sběr, třídění a další jiné manipulace s daty, s cílem tato data vylepšit nebo z nich dostat smysluplnou informaci. Cestou je použití správných nástrojů, které jsou pro tuto úlohu přizpůsobeny. [17]

3.1 Vývoj nástrojů pro práci s daty

3.1.1 Tabulkové procesory

Tabulkový procesor je počítačový program zobrazující data ve dvojrozměrné tabulce a obsahující vzorce, které pracují s těmito daty. Původní účel tabulkového procesoru je zpracování účetních dat, ale v současnosti má i jiná využití, používá se například pro práci s kvantitativními výsledky vědeckých experimentů nebo pro matematické výpočty. [18][19]

Prvním tabulkovým procesorem byl VisiCalc, který byl vyvinut v roce 1979 a dal se použít na osobním počítači. V devadesátých letech se nejpoužívanějším tabulkovým procesorem stal program Excel společnosti Microsoft. Excel fungoval v operačním systému Windows, na který ostatní tabulkové procesory přešly až se zpožděním. Důvodem, proč je Excel tak rozšířený, nejsou ani tak jeho vlastnosti, jako spíše snaha společností mít tabulkový procesor kompatibilní s dalšími programy (textovým editorem, databázovými programy). [18][19]

Výhody tabulkových procesorů:

- **Formáty** – Tabulkové procesory generují soubory v různých formátech, které jsou mezi sebou více či méně kompatibilní. Univerzálním formátem je CSV.
- **Tvorba databází** - v tabulkových procesorech lze vytvářet několik tabulek a navzájem je propojit vztahy. Podporují tedy tvorbu jednodušších databází.
- **Import z databázových programů** - Tabulkové procesory podporují import dat z různých databází a jejich následné zpracování.
- **Funkce, vzorce, filtry**- Tabulkové procesory umožňují použití funkcí různých druhů. Matematické funkce umožňují počítat s daty pomocí matematických operací. Textové funkce umožňují pracovat s textovými hodnotami. Vyhledávací funkce jsou užitečné k vyhledávání v rozsáhlých tabulkách. Informační funkce se používají pro zjišťování informací o hodnotách uvedených v buňkách. Většina z nich vrací logickou hodnotu. Logické funkce provádějí logické testování hodnot a jejich výsledkem jsou logické hodnoty pravda nebo nepravda.
- **Vizualizace dat** - velkou výhodou tabulkových procesorů je snadná vizualizace dat. Výsledná data lze nejen zobrazovat ve formě tabulek, kdy zvýrazníte požadované hodnoty (i za využití podmíněného formátování, minigrafu), ale také přehledně zobrazit v grafech.
- **Makra** - Makra slouží k provádění repetitivních úkonů, které je možné zautomatizovat. V Excelu se makra vytváří pomocí programovacího jazyka VBA (Visual Basic for Applications). Buď je možné kód přímo napsat, nebo si pomocí speciální funkcionality "Záznam makra" nahrát sled úkonů, které má makro vykonávat.

[18][19]

3.1.2 Databáze

Databáze je systém souborů s pevnou strukturou záznamů. Tyto soubory jsou mezi sebou navzájem propojeny pomocí klíčů. V širším smyslu jsou součástí databáze i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento software se v české odborné literatuře nazývá systém řízení báze dat (SŘBD). Běžně se označením databáze – v závislosti na kontextu – myslí jak uložená data, tak i software (SŘBD). [20]

Předchůdcem databází byly papírové kartotéky. Dalším krokem bylo převedení zpracování dat na stroje. Paměťovým médiem byl děrný štítek a zpracování sebraných informací probíhalo na elektromechanických strojích. Elektromechanické stroje se využívaly pro účely zpracování dat další půlstoletí. Velkým impulsem pro další rozvoj databází byl překotný vývoj počítačů v padesátých letech 20. století. Ukázalo se, že původně univerzální používání strojového kódu procesorů je (nejen) pro databázové úlohy neefektivní, a proto se objevil požadavek na vyšší jazyk pro zpracování dat. Začaly vznikat první síťové SŘBD na sálových počítačích. V roce 1970 začínají zveřejněním článku E. F. Codd první relační databáze, které pohlíží na data jako na tabulky. Kolem roku 1974 se vyvíjí první verze dotazovacího jazyka SQL. Vývoj této technologie po 10 letech přinesl výkonově použitelné systémy, srovnatelné se síťovými a hierarchickými databázemi. [20]

Nástroje pro práci s databázemi: Microsoft Access, MySQL, PostgreSQL, Microsoft SQL Server [20]

3.1.3 National Instruments DIAdem

DIAdem je software firmy National Instruments zabývající se vyhledáváním, a off-line analýzou naměřených dat, jejich zobrazením a tvorbou protokolů. Obsahuje sedm modulů, pro efektivní a rychlé provádění analýzy. Moduly a jim přidružené funkce jsou sdruženy do záložek, mezi kterými existuje spojovací tunel zvaný DataPortal. Součástí je také DataFinder, ten slouží k přípravě dat na disku a síti pro rychlé vyhledávání formou indexace, podobně jako to dělají internetové vyhledávače. Tento proces značně zvyšuje rychlost vyhledávání. [21]

Souhrn modulů DIAdem:

- **Navigator** - Modul, jenž na základě indexů vyhledává požadované informace a data, vyhledávání lze soustředit na kteroukoli ze tří již zmíněných úrovní TDM vrstvy.
- **View** - Zde uživatel data z Data Portálu zobrazuje na grafech, tabulkách či videích.
- **Analysis** - Tento modul slouží k analýze dat. Lze aplikovat různé matematické funkce a operace. Z výsledků lze vytvořit nové soubory.
- **Report** - slouží k vytváření protokolů.
- **DAC** - Součástí je velice užitečný DIAdem DAC, který slouží k měření dat. Celý proces od akvizice, přes ukládání až po zobrazení se skládá z funkčních bloků, které dohromady tvoří blokový diagram.
- **Visual** - V této záložce se zobrazují měřená data z měřicího řetězce modulu DAC. Celkově tento modul slouží jako styčný bod měřicího řetězce a uživatele, kde uživatel pomocí tlačítek vyvolává události, které ovlivňují samotné měření. Například požadavek pro uložení dat.
- **Script** - Pro automatizaci práce s výše zmíněnými moduly lze psát algoritmy, které vše značně urychlují a umožňují stejný proces opakovat pro jiná data.

Velkou výhodou tohoto přístupu k datům je urychlení vyhledávání a zpracování velkého množství dat najednou. Nevýhodou tohoto přístupu může být nedostatečné těžení užitečných informací z datové sady. Stále je důležité mít k dispozici dobré znalosti z oblasti analýzy dat, jelikož celý tento koncept je pouze prostředek k managementu s daty. [21]

3.2 Moderní nástroje pro práci s daty

Moderní metody zpracování dat se vyznačují především technikou strojového učení. To označuje proces, při kterém počítač ve vstupních datech rozpoznává vzory (patterns) odpovídající sledovanému jevu.

Důležitým prvkem při strojovém učení je tzv. neuronová síť, která se podobá neuronové síti v lidském mozku. Tu je možné si představit jako složitou matematickou funkci, na jejímž vstupu jsou zpracovávaná data. Ve zjednodušené podobě, mezi jednotlivými neurony v síti jsou datová spojení, s nastavitelnými koeficienty – na nich záleží, jak se vstupní data promítnou do výstupních dat.

Strojové učení „nenaučené“ sítě pak probíhá tak, že se na vstup přivede jeden ze vzorků dat a síť pro něj vrátí jistou výstupní hodnotu. Pokud tato není správná, změní se nastavení koeficientů tak, aby se výsledek opravil na správnou hodnotu, a takto se postupuje pro všechny vstupní vzorky. V ideálním případě je výsledkem naučená síť, nahrazující konvenční počítačový algoritmus, plnící zadaný úkol.

3.2.1 Python

Velmi populárním nástrojem v oblasti Data science je programovací jazyk Python. Je to jazyk interpretovaný, multiplatformní a univerzálně použitelný. V Data Science je užíván pro přípravu a vizualizaci dat, samotné algoritmy provádějící strojové učení jsou implementovány (s ohledem na rychlost) v jiných jazycích – typicky C++.

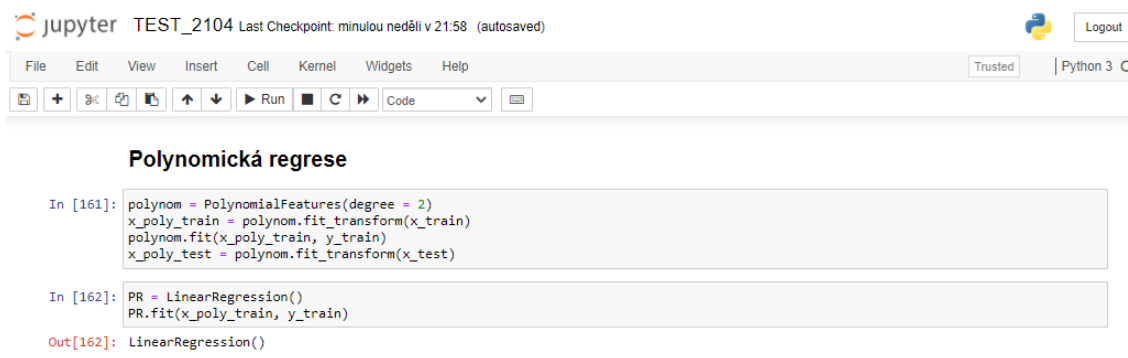
Ačkoli jazyk Python, ani jeho standardní knihovna, funkce specializované pro Data Science neobsahují, plní tuto úlohu knihovny třetích stran. Následuje krátký popis nejčastěji užívaných knihoven.

- **NumPy** – knihovna NumPy přináší v Pythonu zejména chybějící podporu pro vícedimenzionální pole a rychlé výpočty s nimi (díky implementaci výpočtů v kompilovaných jazycích). Dále obsahuje funkce pro provádění FFT, výpočty v oblasti lineární algebry, a usnadňuje integraci knihoven psaných v C++/Fortranu.
- **Pandas** – celým názvem Python Data Analysis Library. Pandas je knihovna usnadňující práci s daty (podporuje data v různých formátech - např. Excel, SQL tabulce, časové průběhy naměřených veličin typické pro různá měření...), zaměřená s ohledem na podoby dat vyskytujících se v běžné praxi. Obsahuje mnoho předpřipravených datových struktur, s optimalizovanými algoritmy pro běžné úkony.
- **Scikit-Learn** – Scikit-Learn je knihovna pro strojové učení, užívající ke své činnosti mj. knihovny NumPy a Matplotlib. Je navržena se záměrem být co nejuniverzálnější a co nejjednodušší použitelná pro kohokoli, a to včetně komerčních užití.
- **Keras** – Keras je knihovna pro deep learning. Sama neimplementuje algoritmy strojového učení, ale využívá jiné – např. TensorFlow. Jejím záměrem je umožnit rychlé experimentování. Umožňuje mj. vytváření a úpravu modelů (neuronových sítí) a jejich běh na CPU i GPU.

- **Matplotlib** – knihovna Matplotlib slouží k vykreslování různých typů grafů vhodných k publikaci, např. klasických křivkových, polárních, sloupcových, koláčových atd. Je možno velmi podrobně specifikovat výslednou podobu grafu.
- **Seaborn** – Seaborn je knihovna pro vizualizaci dat využívající knihovnu Matplotlib. Na rozdíl od knihovny Matplotlib je však zaměřena na snadné použití pro datové sady – zahrnuje v sobě operace nutné k „předpřípravě“ dat pro knihovnu Matplotlib. Výsledkem je usnadnění vizualizace dat – oproti knihovně Matplotlib stačí mnohdy jediné volání funkce.

3.2.2 Jupyter Notebook

Je to IDE (Integrated Development Environment), tj. prostředí pro vývoj softwaru. Jedná se o jednoduché a velmi interaktivní vývojové prostředí, běžící v jakémkoli internetovém prohlížeči. Uživatel tak může pracovat ve známém prostředí, což přispívá k jeho komfortu. Programy v Pythonu lze psát v jakémkoli textovém editoru a spouštět z příkazové řádky, ale užití Jupyteru programování usnadňuje tím, že lze kód rozdělit do buněk, které lze spouštět nezávisle na ostatních, což usnadňuje ladění vytvářeného programu. Jupyter je webová aplikace s otevřeným zdrojovým kódem, která umožňuje vytvářet a sdílet dokumenty, které obsahují živý kód, rovnice, vizualizace a narativní text. Mezi hlavní výhody patří podpora široké škály programovacích jazyků jako jsou Python, R nebo Scala.



The screenshot shows the Jupyter Notebook interface. At the top, there's a header with the Jupyter logo, the name of the notebook 'TEST_2104', and the last checkpoint information 'Last Checkpoint: minulou neděli v 21:58 (autosaved)'. There's also a 'Logout' button. Below the header is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. To the right of the menu bar, there's a 'Trusted' status indicator and the Python version 'Python 3'. Below the menu bar is a toolbar with various icons for file operations, cell navigation, and execution. The main area of the notebook is titled 'Polynomial regression'. It contains two code cells. The first cell, labeled 'In [161]:', contains the following code:

```
polynom = PolynomialFeatures(degree = 2)
x_poly_train = polynom.fit_transform(x_train)
polynom.fit(x_poly_train, y_train)
x_poly_test = polynom.fit_transform(x_test)
```

. The second cell, labeled 'In [162]:', contains the following code:

```
PR = LinearRegression()
PR.fit(x_poly_train, y_train)
```

. Below the second cell, the output is shown: 'Out[162]: LinearRegression()'.

Obrázek 3.1: Ukázka zdrojového kódu v prostředí Jupyter Notebook

Na obrázku výše je ukázka zdrojového kódu Python v prostředí Jupyter Notebook využívající knihovny pro strojové učení. Tato problematika bude podrobněji probrána v kapitole 4.

4 Případová studie

4.1 Prediktivní údržba

Prediktivní údržba představuje analytický přístup, který umožňuje předpovídat, kdy dojde k selhání výrobních zařízení a předcházet jim včasnou a efektivní údržbou. Může však jít ještě dále a poskytovat informace, na základě kterých je možné zařízení postupně „ladit“ a sklony k poruchovosti snižovat až téměř eliminovat. Nezanedbatelnou součástí preventivní údržby je také vyhodnocování faktorů ovlivňujících kvalitu výrobků a zavádění opatření směřující k její optimalizaci.

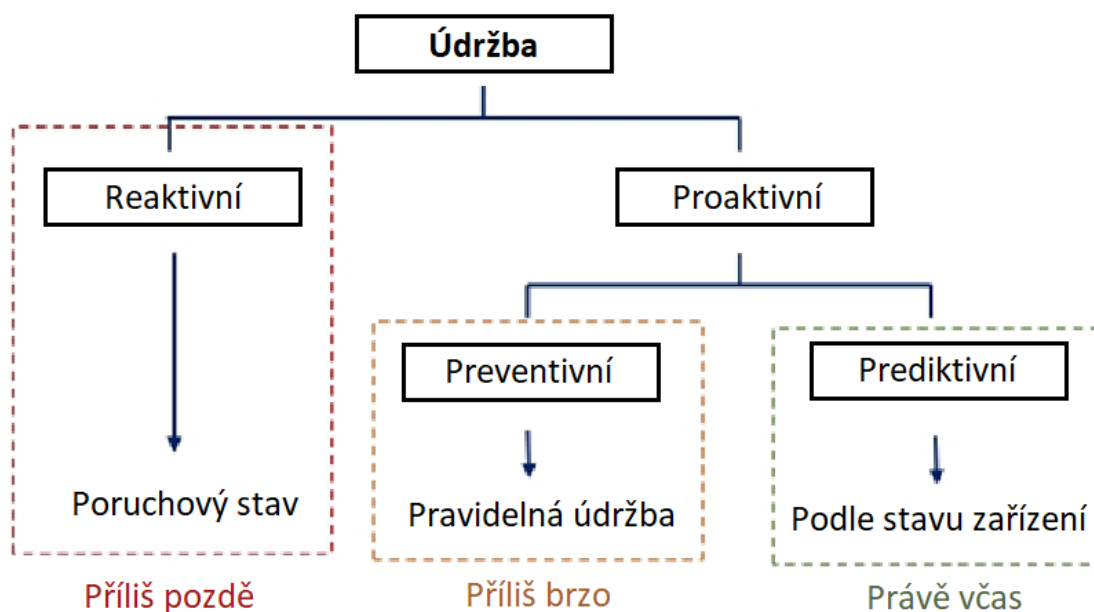
Mezi hlavní funkce informačního systému údržby patří zpracování, vizualizace, analýza a archivace měřených dat. Automatické generování alarmů při překročení nastavených mezních hodnot a hlášení ve formě SMS, e-mailů a generování požadavků na údržbu nebo pracovních příkazů. [22]

4.1.1 Vývoj průmyslové údržby

Reaktivní údržba – nejstarším typem údržby je údržba reaktivní. Její princip spočívá v tom, že zásah je proveden až v okamžiku, kdy dojde k havárii nebo poškození daného systému. Havárii tedy nemůžeme vyloučit. Odstranění takové havárie je pak většinou velmi nákladné, především s ohledem na neočekávané výpadky výroby. S trochou nadsázky se dá takový typ údržby označit jako „hašení požáru“, což v některých případech může platit i doslovně. [23]

Preventivní údržba – pokročilejším typem údržby je údržba preventivní. Jak vyplývá z názvu, její podstatou je prevence, tzn. preventivní kontroly a výměna kritických komponentů v pravidelných časových intervalech, bez ohledu na jejich stav. Tento typ údržby je technologicky náročnější než údržba reaktivní, dokáže však výrazně snížit riziko havárie. Po finanční stránce mohou náklady na preventivní údržbu být nižší než náklady na havárii. [23]

Prediktivní údržba – nejmodernějším typem údržby je údržba prediktivní (predikce = předpověď). Principem tohoto typu údržby je předpovídat vývoj stavu výrobních zařízení a včas odhalit potencionální problém či poruchu. Cílem toho typu organizace údržby je mít o udržovaném zařízení co nejvíce informací, a to průběžně tak, aby byl jasný vývoj změny vlastností, parametrů atd. v čase. Pak je možné opravu dané části zařízení provést dříve, než dojde k jeho poškození či havárii, ve zvoleném okamžiku a s již předem objednanými díly. Odstavení výroby je tak plánované, je možné provést výrobu na „sklad“ a její výpadek má minimální dopady na následující logistiku zboží a prodeje. [23]



Obrázek 4.1: diagram vývoje průmyslové údržby [24]

4.1.2 Využití prediktivní údržby

Maximalizace produktivity aktiv - Mezi aktiva organizace mohou patřit výrobní stroje, koncová zařízení (uživatelské spotřebiče, prodejní automaty, strojní zařízení, ...) nebo budovy. [25]

Řízení a zvyšování kvality - Maximalizace kvality je pro výrobu poměrně složitá a nákladná. Pokud dojde k poruše výrobní linky, musí ji (v oborech, kde to je možné) po určitou dobu nahradit manuální práce. Kvalitu v tomto případě ovlivňuje čistě lidský faktor, který je standardně daleko méně spolehlivý než stroje. Klíčem ke zvýšení kvality je včasná identifikace problému díky zapojení kontrolních mechanismů přímo do výrobního procesu místo kontroly a vyřazování vadných produktů až na konci výrobního cyklu, kdy už je na nápravu pozdě. [25]

Optimalizace souvisejících procesů - Prediktivní údržba se sice zaměřuje primárně na výrobní a provozní zařízení, ale souvisí také s dalšími procesy v organizaci, například s řízením kvality, zásob, záruční politikou, obchodní a výrobní činnostmi, financemi, plánováním, rozpočtováním a řízením lidských zdrojů. [25]

Snížení nákladů na zásoby a jejich řízení - Dlouhodobější poruchy zařízení negativně ovlivňují produkci – pokud nefunguje výrobní linka, neprodukuje firma výrobky. Zároveň se začínají hromadit vstupní komponenty a suroviny, což ve skladech způsobuje problémy a nepřímo dále zvyšuje náklady. Při klasické preventivní údržbě je ve skladech také potřeba držet dostatečné zásoby náhradních dílů, aby údržbový tým dokázal porouchaná zařízení co nejdříve opravit. [25]

Zlepšení finančních procesů - Díky kombinaci nástrojů pro prediktivní údržbu s nástroji pro plánování, reporting a analýzu dat je dále možné kombinovat data o poptávce, realizovaných objednávkách, kapacitě, stavu zařízení a zásobách s finančním plánem a klíčovými ukazateli výkonnosti a podporovat tak kvalifikovaná rozhodnutí založená na informacích místo na pocitech. [25]

Snížení počtu uplatňovaných reklamací - Pokud porucha výrobního zařízení způsobí, že se ke spotřebiteli dostanou vadné výrobky, zvyšují se počty reklamací a náklady na jejich řešení. Prediktivní údržba dokáže včas určit souvislost reklamací s problémy na výrobních zařízeních a umožní omezit nákladná reklamační řízení poskytnutím potřebného servisu dříve, než se u většiny zákazníků projeví. [25]

4.2 Nalezení vhodných datových sad

Klíčovým předpokladem inteligentních systémů údržby je schopnost předpovědět zbývající životnost jeho komponent, tj. prognostika. Vývoj prognostických modelů založených na datech vyžaduje datové sady s trajektoriemi run-to-failure tj. měření až do bodu selhání. Velké reprezentativní datové sady typu run-to-failure jsou však ve skutečných aplikacích často nedostupné, protože poruchy jsou v mnoha systémech důležitých z hlediska bezpečnosti vzácné. Poruchy kriticky důležitých systémů, mohou způsobit významná ekonomická narušení a mohou mít vysoké náklady na odstranění poruchy a ztrátou ve výrobě. Možnou cestou je simulace daného systému např. pomocí digitálního dvojčete, kde můžeme tento systém testovat až do bodu selhání. Samozřejmě v praxi se tento bod může lišit, a proto je nutné počítat s odchylkou.

V této práci jsem se rozhodl nalézt už hotovou datovou sadu, na které lze aplikovat prediktivní údržbu. Problémem je, že málokterá společnost chce zveřejnit data ze svých výrobních procesů, aby např. nebyla zneužita konkurencí vyzrazením jejich know-how. Avšak i přes to lze na internetu nalézt mnoho kvalitních datových sad, které lze pro tento účel využít. K nalezení vhodné datové sady lze mimo jiné využít např. internetové vyhledávače datových sad a pomocí klíčových slov (např. Predictive maintenance) nalézt datovou sadu, která splňuje předem stanovená kritéria.

Příklady vyhledávačů datových sad:

- **Google Dataset Search** [45] - Tento vyhledávač datových sad byl spuštěn v roce 2018, je jako standardní vyhledávač Google, ale výhradně pro data.
- **Kaggle** [46]- Stejně jako Vyhledávání datových sad Google nabízí Kaggle agregované datové sady, ale je to spíše komunitní centrum než vyhledávač. Společnost Kaggle zahájila v roce 2010 řadu soutěží v oblasti strojového učení, které následně vyřešily problémy jako NASA a Ford.
- **Data.Gov** [47] - V roce 2015 vláda USA zveřejnila všechny své údaje. V tomto vyhledávači jsou data vlády zveřejňována na podporu výzkumu a vývoje ve vědeckých komunitách.

Další možností je hledání na relevantních internetových fórech jako např. reddit [26], kde uživatelé mohou poradit nebo přímo dodat potřebná data. Tato Data Science komunita mě nasměrovala na stránky americké NASA, konkrétně na prognostické datové úložiště [27], tj. sbírka datových sad, které byly darovány různými univerzitami, agenturami nebo společnostmi. Úložiště dat se zaměřuje výhradně na sady prognostických dat, tj. sady dat, které lze použít pro vývoj prognostických algoritmů. Většinou se jedná o časové řady dat z nějakého nominálního stavu do stavu selhání.

4.3 Výběr vhodné konkrétní datové sady

Pomocí výše uvedených zdrojů jsem našel několik datových sad, na kterých by bylo vhodné aplikovat princip prediktivní údržby. Seznam mnou nalezených datových sad:

Battery Data Set [28] – Tato datová sada je výsledkem pokusů s Li-Ion bateriemi. Obsahuje nabíjení a vybíjení při různých teplotách. Zaznamenává impedanci jako kritérium poškození. Soubor dat poskytl Prognostics CoE v NASA Ames.

- Formát dat: .mat
- Úroveň dokumentace: dobrá
- Velikost datové sady: 200 MB
- Referenční řešení pro validaci postupu: ano

Bearing Data Set [29] - tato datová sada je výsledkem pokusů s ložisky. Tyto data poskytlo Centrum pro inteligentní systémy údržby (IMS), University of Cincinnati.

- Formát dat: .csv
- Úroveň dokumentace: špatná
- Velikost datové sady: 1,07 GB
- Referenční řešení pro validaci postupu: ano

Randomized Battery Usage Data Set [30]- Tyto data jsou výsledkem pokusů, kde jsou baterie nepřetržitě cyklovány náhodně generovanými aktuálními profily. Referenční cykly nabíjení a vybíjení se také provádějí po pevném intervalu náhodného použití, aby se poskytly referenční měřítka pro stav baterie.

- Formát dat: žádný
- Úroveň dokumentace: dobrá
- Velikost datové sady: 1 GB
- Referenční řešení pro validaci postupu: ano

XJTU-SY_Bearing_Datasets [31]- Datová sada obsahuje kompletní data run-to-failure 15 valivých ložisek, která byla získána provedením mnoha experimentů s urychlenou degradací.

- Formát dat: .csv
- Úroveň dokumentace: špatná
- Velikost datové sady: 1,38 GB
- Referenční řešení pro validaci postupu: ano

Turbofan Engine Degradation Simulation Data Set [32]- Simulace degradace motoru byla provedena pomocí C-MAPSS. Čtyři různé sady byly simulované za různých kombinací provozních podmínek a poruchových režimů. Zaznamenává několik kanálů senzoru, aby charakterizoval vývoj poruchy. Soubor dat poskytl Prognostics CoE v NASA Ames.

- Formát dat: .txt
- Úroveň dokumentace: dobrá
- Velikost datové sady: 12 MB
- Referenční řešení pro validaci postupu: ano

Turbofan Engine Degradation Simulation Data Set-2 [33]- Tato datová sada je novější verzí předchozí datové sady. Soubor dat poskytuje Prognostics CoE v NASA Ames ve spolupráci s ETH Zurich a PARC.

- Formát dat: .csv
- Úroveň dokumentace: dobrá
- Velikost datové sady: 14,7 GB
- Referenční řešení pro validaci postupu: ne

Tabulka 4.1: porovnání datových sad

název	formát dat	dokumentace	ref. řešení	velikost
Battery Data Set	.mat	dobrá	ano	200 MB
Bearing Data Set	.csv	špatná	ano	1.07 GB
Randomized Battery Usage Data Set	žádný	dobrá	ano	1GB
XJTU-SY_Bearing_Datasets	.csv	žádná	ano	1,38GB
Turbofan Engine Degr. Sim. Data Set	.txt	dobrá	ano	12 MB
Turbofan Engine Deg. Sim. Data Set-2	.csv	dobrá	ne	14,7 GB

Při výběru datové sady pro bakalářskou práci jsem vybíral podle čtyř hlavních parametrů, tj. formát dat, úroveň dokumentace, zda existuje referenční řešení pro validaci výsledků a velikost datové sady. Z tohoto výběru jsem nejdříve vyřadil datové sady se špatnou nebo chybějící dokumentací, tedy Bearing Data Set [29] a XJTU-SY_Bearing_Datasets [31], protože bez řádné dokumentace je téměř nemožné se v datech orientovat a porozumět jim. Dále jsem z výběru vyřadil datové sady, ke kterým se mi nepodařilo nalézt vypracované referenční řešení pro ověření mých výsledků, tj. Turbofan Engine Deg. Sim. Data Set-2 [33]. Nakonec jsem vyřadil z výběru i Randomized Battery Usage Data Set [30], protože data v něm nejsou strukturovaně uložena.

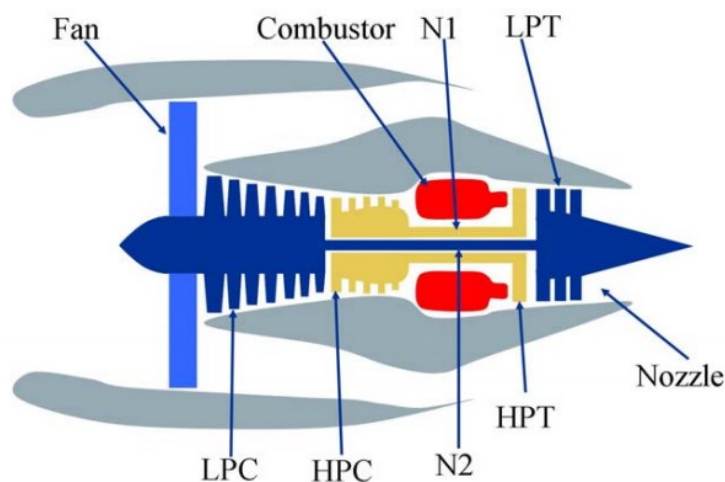
Ve výběru zbyly dvě ideální datové sady tj. Battery Data Set [28] a Turbofan Engine Degr. Sim. Data Set [32]. Z těchto dvou jsem se rozhodl použít Turbofan Engine Degr. Sim. Data Set [32] z důvodu, že je velmi rozšířený v data science komunitě a je k němu zpracováno mnoho referenčních řešení.

4.4 Popis struktury vybrané datové sady a cíle její analýzy

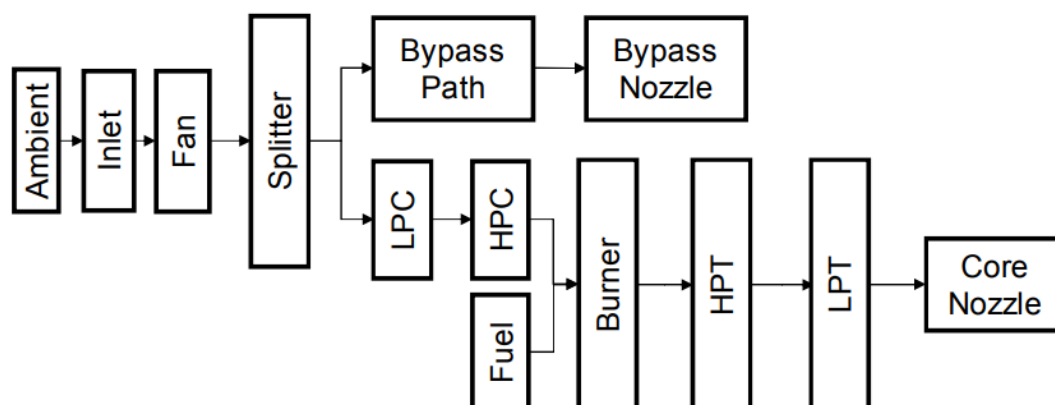
C-MAPSS je nástroj pro simulaci realistického velkého komerčního turboventilátoru. Software je kódován v prostředí MATLAB® a Simulink® a obsahuje řadu upravitelných vstupních parametrů, které uživatelům umožňují zadávat konkrétní hodnoty podle jeho vlastní volby týkající se provozního profilu, regulátorů uzavřené smyčky, podmínek prostředí atd. C -MAPSS simuluje model motoru třídy tahu 90 000 lb a balíček obsahuje atmosférický model schopný simulovat operace v nadmořských výškách od hladiny moře do 40 000 ft, Machových číslech od 0 do 0,90 a teploty na úrovni moře od – 60 do 103 ° F. Součástí balení je také systém řízení výkonu, který umožňuje provozování motoru v širokém rozsahu úrovní tahu v celém rozsahu letových podmínek. [32]

Integrovaný řídicí systém se dále skládá z regulátoru otáček ventilátoru a sady regulátorů a omezovačů. Posledně jmenované zahrnují tři regulátory vysokých limitů, které brání motoru v překročení konstrukčních limitů pro otáčky jádra, poměr tlaku motoru a výstupní teplotu vysokotlaké turbíny (HPT); omezovač, který zabráňuje příliš nízkému statickému tlaku na výstupu z vysokotlakého kompresoru (HPC); a omezovač zrychlení a zpomalení pro rychlost jádra. Komplexní logická struktura

integruje tyto komponenty řídicího systému podobným způsobem, jaký se používá v řadičích skutečných motorů, takže nedochází k problémům s navíjením integrátoru. Kromě toho jsou všechny zisky pro regulátor otáček ventilátoru a čtyři regulátory limitu naplánovány tak, aby regulátor a regulátory fungovaly podle očekávání v celém rozsahu letových podmínek a úrovní výkonu. Schéma motoru na obrázku 1 ukazuje hlavní prvky modelu motoru a vývojový diagram na obrázku 2 ukazuje, jak jsou v simulaci sestaveny různé podprogramy. [32]



Obrázek 4.2: zjednodušený diagram motoru simulovaného v C-MAPSS [32]



Obrázek 4.3: Rozložení a propojení modulů tak, jak jsou simulovány v C-MAPSS [32]

Tabulka 4.2: rozbor datových sad

Datová sada	FD001	FD002	FD003	FD004
Počet motorů	100	260	100	249
Trénovací vzorky	20 631	53 579	24 270	61 249
Testování vzorky	13 096	33 991	16 596	41214
Počet datových sloupců	26	26	26	26
Průměrná délka života (cykly)	206	137	247	245
Operační podmínky	1	6	1	6

Datové sady se skládají z několika časových řad s více proměnnými. Každá sada dat je dále rozdělena na tréninkové a testovací podmnožiny. Každá časová řada pochází z jiného motoru - tj. Údaje lze považovat za údaje z flotily motorů stejného typu. Každý motor začíná jiným stupněm počátečního opotřebení a výrobními odchylkami, které jsou uživateli neznámé. Toto opotřebení a variace se považují za normální, tj. Nepovažují se za poruchové stavy. Existují tři provozní nastavení, která mají podstatný vliv na výkon motoru. Tato nastavení jsou také zahrnuta v datech. Data jsou kontaminována šumem snímače.

Motor pracuje normálně na začátku každé časové řady a během uložené série dat dojde k chybě. V tréninkové sadě porucha narůstá až do selhání systému. V testovací sadě časová řada končí nějakou dobu před selháním systému. Cílem navržené analýzy dat je předpovědět počet zbývajících provozních cyklů před selháním v testovací sadě, tj. počet provozních cyklů po posledním cyklu, kdy bude motor pokračovat v provozu. Je zde k dispozici i vektor skutečných hodnot Remaining Useful Life (RUL) pro data testu.

Data jsou poskytována jako textový soubor komprimovaný pomocí zipu s 26 sloupci čísel oddělených mezerami. Každý řádek představuje snímek dat pořízených během jediného provozního cyklu, každý sloupec je jinou proměnnou. Sloupce ve struktuře dat odpovídají:

- 1) číslo motoru
- 2) cyklus
- 3) provozní nastavení 1
- 4) provozní nastavení 2
- 5) provozní nastavení 3
- 6) senzorové měření 1
- 7) senzorové měření 2
- ...
- 32) senzorové měření 26

Fyzikální jednotky v této datové sadě jsou anglosaské:

°R – Stupeň Réaumura je jednotka teploty pojmenovaná po francouzském přírodovědci Reném Réaumurovi, který ji zavedl roku 1730, ale dnes se již téměř nepoužívá.

Psia – je absolutní libra na čtvereční palec. Je to tlak, který je výsledkem síly jedné libry působící na plochu jednoho čtverečního palce.

lbm / s – je libra za sekundu. Tato jednotka popisuje množství hmotnostního průtoku.

Tabulka 4.3: Popis senzorů v datové sadě [32]

Číslo senzoru	Značka	Popis	Jednotka
1	T2	Celková teplota na vstupu ventilátoru	°R
2	T24	Celková teplota na výstupu LPC	°R
3	T30	Celková teplota na výstupu HPC	°R
4	T50	Celková teplota na výstupu LPT	°R
5	P2	Tlak na vstupu ventilátoru	Psia
6	P15	Celkový tlak v obtokovém potrubí	Psia
7	P30	Celkový tlak na výstupu HPC	Psia
8	Nf	Fyzická rychlost ventilátoru	ot. / min
9	Nc	Rychlost fyzického jádra	ot. / min
10	Epr	Poměr tlaku motoru	-
11	Ps30	Statický tlak na výstupu HPC	Psia
12	Phi	Poměr průtoku paliva k Ps30	Pps / psi
13	NRf	Opravená rychlost ventilátoru	ot. / min
14	NRc	Opravená rychlost jádra	ot. / min
15	BPR	Poměr bypassu	-
16	farB	Poměr paliva a vzduchu v hořáku	-
17	hfBleed	Entalpie krvácení	-
18	Nf_dmd	Požadovaná rychlost ventilátoru	ot. / min
19	PCNtR_dmd	Požadovaná opravená rychlost ventilátoru	ot. / min
20	W31	Odvzdušnění chladicí kapaliny HPT	lbm / s
21	W32	Odvzdušnění chladicí kapaliny LPT	lbm / s

Cíl analýzy

Cílem analýzy této datové sady je určit kritické parametry a následně z nich vytvořit model pro monitoring stavu motoru a predikci poruchových stavů. To zahrnuje činnosti jako předběžnou analýzu datové sady, předzpracování dat, identifikace indikátorů stavu, vytvoření potřebných funkcí pro trénování modelu, vizualizaci dat a ověřování správnosti modelů. Dále samotné trénování modelů, včetně jejich parametrizace pro dosažení nejlepšího výsledku. Následuje výběr nejlepšího modelu a srovnání s referenčním řešením, ve kterém porovnám výsledky obou stran a pokusím se identifikovat možné důvody případných rozdílů. Nakonec navrhnu postup případného vylepšení mého řešení.

4.5 Výběr nástroje pro jeho analýzu

Pro svou bakalářskou práci jsem se rozhodl provést analýzu pomocí nástroje Jupyter Notebook, což je webová aplikace s otevřeným zdrojovým kódem, která umožňuje vytvářet a sdílet dokumenty, které obsahují živý kód, rovnice, vizualizace a narativní text. Mezi jeho použití patří: čištění a transformace dat, numerická simulace, statistické modelování, vizualizace dat, strojové učení a mnoho dalšího.

Výhody Jupiteru:

- Volba programovacího jazyku – Jupyter podporuje více než 40 programovacích jazyků, včetně Pythonu, R, Julie a Scaly.
- Snadné sdílení – Notebooky lze sdílet s ostatními pomocí e-mailu, Dropboxu, GitHub a Prohlížeče poznámek Jupyter.
- Interaktivní výstup – Kód může produkovat bohatý interaktivní výstup: HTML, obrázky, videa, LaTeX a vlastní typy MIME.

IDE Jupyter Notebook je velmi oblíbený v Data Science i díky jeho struktuře programování, kde se kódy vytvářejí do oddělených buněk, které lze spouštět nezávisle na ostatních, což usnadňuje ladění vytvářeného programu. Pro toto IDE jsem se rozhodl, protože jsem během studia v něm pracoval a velice rychle jsem si práci v něm oblíbil. Ze stejného důvodu jsem se rozhodl pro použití jazyka Python, který je teď nejvíce používaný jazyk obecně ve strojovém učení, ale i v programování obecně. Navíc podporuje užitečné knihovny funkcí pro strojové učení.

4.6 Popis jednotlivých kroků analýzy

4.6.1 Import knihoven a načtení dat

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set_style('darkgrid')
import math
#metriky
from sklearn.metrics import mean_squared_error, mean_absolute_error
#modely strojového učení
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import RandomForestRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.svm import SVR
from sklearn.ensemble import GradientBoostingRegressor
#pro normalizování dat
from sklearn.preprocessing import MinMaxScaler
```

Obrázek 4.4: Kód programu pro načtení knihoven funkcí

Importoval jsem knihovnu Numpy pro práci s více rozměrnými poli, vektory a maticemi. Dále knihovnu Pandas pro manipulaci a analýzu dat. Knihovnu Seaborn pro vizualizaci dat a knihovnu Math matematické funkce.

Dále jsem importoval metriky pro ověření správnosti predikce MSE (Mean Squared Error) a MAE (Mean Absolute Error) z knihovny sklearn. Poté jsem importoval predikční modely z knihovny sklearn a funkci pro normalizaci dat.

```
#názvy sloupců pro importovaný dataset
col_names = ['ID', 'Cycle', 'OpSet1', 'OpSet2', 'OpSet3', 'SensorMeasure1', 'SensorMeasure2',
             'SensorMeasure3', 'SensorMeasure4', 'SensorMeasure5', 'SensorMeasure6', 'SensorMeasure7',
             'SensorMeasure8', 'SensorMeasure9', 'SensorMeasure10', 'SensorMeasure11', 'SensorMeasure12',
             'SensorMeasure13', 'SensorMeasure14', 'SensorMeasure15', 'SensorMeasure16', 'SensorMeasure17',
             'SensorMeasure18', 'SensorMeasure19', 'SensorMeasure20', 'SensorMeasure21']

#import trénovacího a testovacího datasetu ze souborů a následné přiřazení jmen sloupců
train = pd.read_csv('CMAPSSData/train_FD001.txt', sep='\s+', header = None, names=col_names)
test = pd.read_csv('CMAPSSData/test_FD001.txt', sep='\s+', header = None, names=col_names)
```

Obrázek 4.5: Kód programu pro načtení dat

Dále jsem načetl data ze souborů a sloupcům přiřadil jména.

4.6.2 Předzpracování dat

Předběžné zpracování dat je často nutné k převodu dat do formy, ze které lze snadno extrahovat indikátory stavu. Předzpracování dat zahrnuje jednoduché techniky, jako je odstraňování odlehlých hodnot a chybějících hodnot, a pokročilé techniky zpracování signálu. [30]

```
#přidání maximálního počtu cyklů pro každý motor do trénovacího datasetu
max_cycles = train.groupby(['ID'], sort=False)['Cycle'].max().reset_index().rename(columns={'Cycle':'MaxCycleID'})
train = pd.merge(train, max_cycles, how='inner', on='ID')
#přidání cílové hodnoty "RUL" pro trénování modelu do trénovacího datasetu
train['RUL'] = train['MaxCycleID'] - train['Cycle']
#výpis trénovacího datasetu
train
```

Obrázek 4.6: Kód programu pro vytvoření cílové proměnné

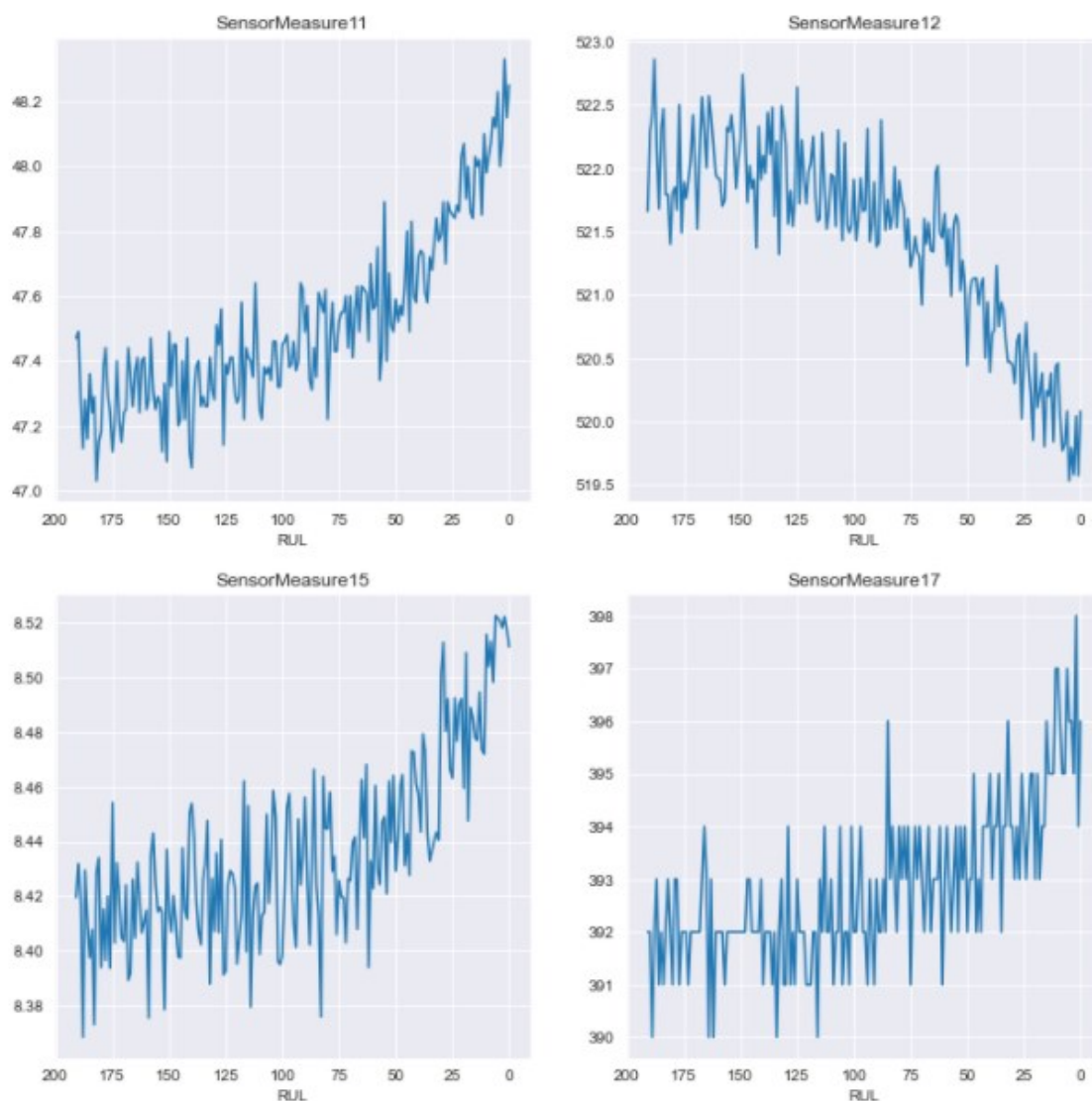
V tomto kroku jsem vytvořil pomocnou proměnou "MaxCycleID", díky které jsem vypočetl cílovou proměnou "RUL", která reprezentuje zbývající životnost motoru v hodinách. Cílová proměnná znamená proměnou, kterou se následně budu snažit předpovědět s co největší přesností. Tuto operaci jsem provedl jak u trénovací datové sady, tak u té testovací.

	ID	Cycle	OpSet1	OpSet2	OpSet3	SensorMeasure1	SensorMeasure2	...	SensorMeasure21	MaxCycleID	RUL
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	...	23.4190	192	191
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	...	23.4236	192	190
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	...	23.3442	192	189
3	1	4	0.0007	0.0000	100.0	518.67	642.35	...	23.3739	192	188
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	...	23.4044	192	187
...
20626	100	196	-0.0004	-0.0003	100.0	518.67	643.49	...	22.9735	200	4
20627	100	197	-0.0016	-0.0005	100.0	518.67	643.54	...	23.1594	200	3
20628	100	198	0.0004	0.0000	100.0	518.67	643.42	...	22.9333	200	2
20629	100	199	-0.0011	0.0003	100.0	518.67	643.23	...	23.0640	200	1
20630	100	200	-0.0032	-0.0005	100.0	518.67	643.85	...	23.0522	200	0

20631 rows x 28 columns

Obrázek 4.7: Výpis dat z trénovací datové sady

V dalším kroku jsem vizualizoval data ze senzorů, abych se pokusil najít informaci, kterou bych mohl využít ke zkvalitnění dat, před trénováním jednotlivých modelů. Na obrázku níže je ukázka z motoru číslo 1 z trénovací datové sady, kde jsou grafy ze čtyř senzorů, které zobrazují výstupní hodnoty v závislosti na RUL. Z grafů lze vypožorovat trend, který se opakuje nejen u ostatních senzorů, ale i u ostatních motorů a to ten, že výstupní hodnoty se nejdříve pohybují kolem konstanty, a následně poblíž RUL = 125 se začínají hodnoty rapidně měnit. Proto jsem vytvořil funkci, která omezuje maximální hodnotu RUL na 125 a díky tomu bude možné predikovat zbývající životnost motoru. Tato operace se provede ve funkci `load_data()`, kterou podrobněji popíšu níže.



Obrázek 4.8: Ukázky dat ze senzorů vůči zbývající životnosti v grafech

Nakonec jsem vytvořil funkci `load_data()`, která rozdělí datové sady na podmnožiny, které budou na vstupu trénování modelů a při ověřování správnosti modelů. V této funkci je také provedena normalizace dat, která je důležitá pro určité typy modelů (např. `RandomForestRegressor`, `Neural Networks`), které jsou velmi náchylné ke špatnému škálování dat.

```

#funkce pro rozdělení dat pro vstup modelů
#ve funkci dojde ke škálování dat
def load_data():
    min_max_scaler = MinMaxScaler()

    x_train = train[features]
    train[features] = min_max_scaler.fit_transform(x_train)
    y_train = train.RUL

    x_test = test[features]
    test[features] = min_max_scaler.fit_transform(x_test)
    y_test = test.RUL

#cílová proměnná RUL je nastavena na max. hodnotu 150
y_train = y_train.clip(upper=125)
y_test = y_test.clip(upper=125)

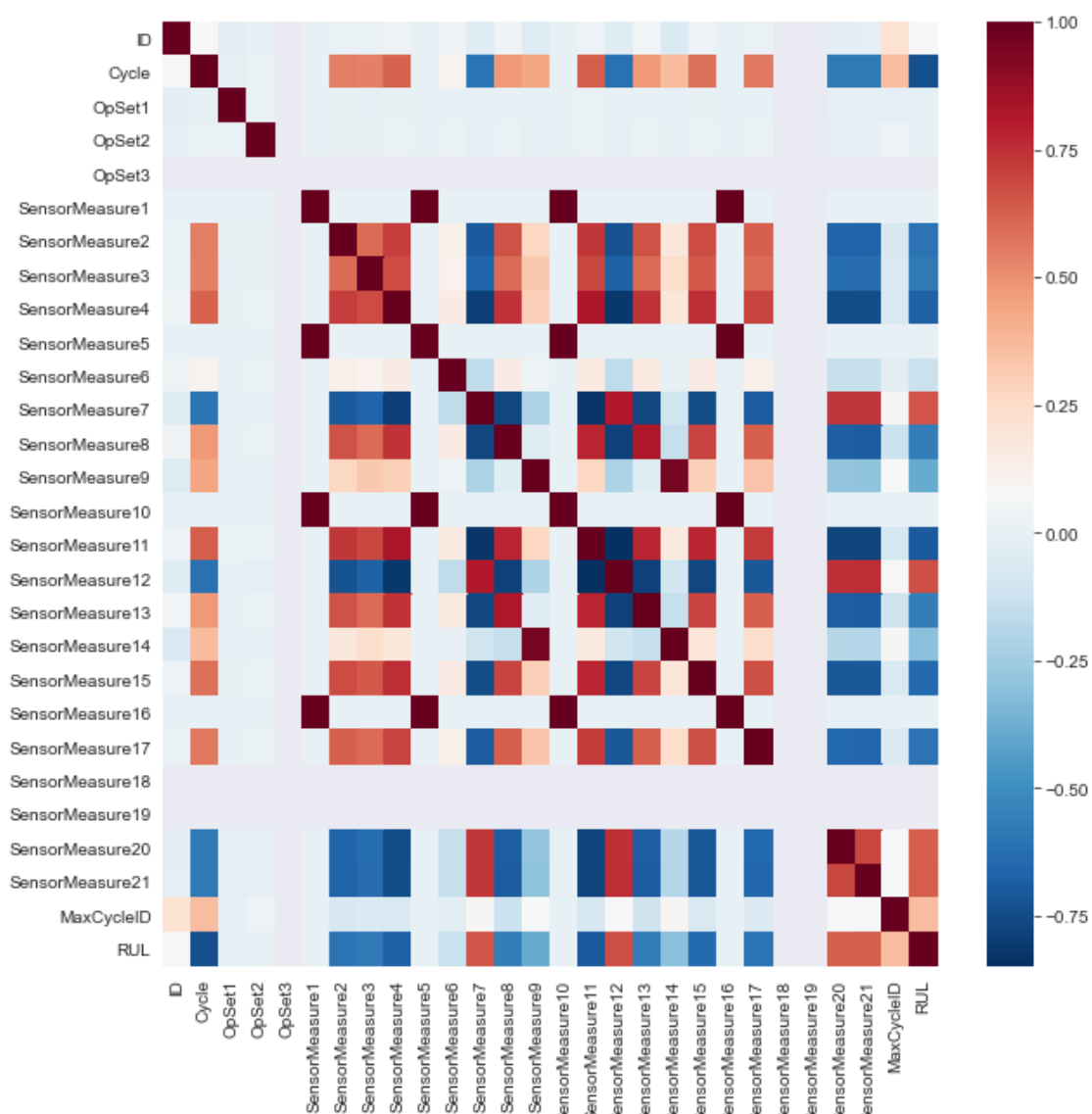
```

Obrázek 4.9: kód programové funkce `load_data()`

4.6.3 Identifikování indikátorů stavu

Klíčovým krokem ve vývoji algoritmu prediktivní údržby je identifikace indikátorů stavu, identifikací funkcí v datech systému, jejichž chování se mění předvídatelným způsobem v průběhu degradace analyzovaného systému. Indikátorem stavu může být jakákoli funkce, která je použitelná pro rozlišení normálního a poruchového provozu nebo pro predikci zbývajících životnosti. Užitečný indikátor stavu seskupuje podobný stav systému a odlišuje poruchový stav. [30]

Pro identifikaci vhodného indikátoru stavu jsem se rozhodl použít korelační funkci, jejíž výstup je matice hodnot, které reprezentují to, jak se proměnné navzájem v datové sadě ovlivňují. Pro lepší pochopení jsem tuto funkci nechal graficky znázornit do korelační mapy.



Obrázek 4.10: Korelační mapa

Na této mapě je důležitý poslední řádek, který ukazuje, jak výrazně ostatní proměnné ovlivňují cílovou proměnnou RUL. Hodnoty blíží se nule nemají skoro žádný vliv na proměnnou RUL, zatímco hodnoty blíží se 1 ovlivňují RUL maximálně v přímé úměrnosti a hodnoty blíží se -1 zase v nepřímé úměrnosti.

Na základě této funkce jsem vybral proměnné, na základě kterých budu předikovat zbývající životnost motorů. Je důležité zmínit, že jsem mimo jiné vyřadil i proměnnou MaxCycleID, protože v reálném použití tento údaj není k dispozici a proto by použití této proměnné bylo nekorektní ($RUL = MaxCycleID - Cycle$).


```
['Cycle',
 'SensorMeasure2',
 'SensorMeasure3',
 'SensorMeasure4',
 'SensorMeasure6',
 'SensorMeasure7',
 'SensorMeasure8',
 'SensorMeasure9',
 'SensorMeasure11',
 'SensorMeasure12',
 'SensorMeasure13',
 'SensorMeasure14',
 'SensorMeasure15',
 'SensorMeasure17',
 'SensorMeasure20',
 'SensorMeasure21']
```

Obrázek 4.11: Indikátory stavu

4.6.4 Výběr verifikačních parametrů

Vytvořil jsem funkci `verification()`, jejíž vstupem jsou predikovaná data RUL a skutečná data RUL. Výstupem této funkce jsou metriky správnosti modelu. V tomto projektu jsem vybral 2 metriky pro ověření správnosti:

- RMSE(root-mean-square-error) - je to míra přesnosti pro porovnání chyb předpovědi různých modelů pro konkrétní datovou sadu. Tuto metriku jsem vybral, protože je používána i v referenčním řešení, tudíž bude sloužit pro porovnání s referenčním řešením. [34]
- MAE(mean-absolute-error) - Střední absolutní chyba. Jde o aritmetický průměr absolutních chyb - rozdílu predikce a skutečné hodnoty. Tuto metriku jsem se rozhodl použít, protože je z ní snadno vidět, jak přesná byla predikce. [35]

```
#funkce pro testování modelů
def verification(y_data,y_pred,train_or_test):
    mse = mean_squared_error(y_data,y_pred)
    rmse = np.sqrt(mse)
    mae = mean_absolute_error(y_data,y_pred)

    if train_or_test:
        print("train set:")
    else:
        print("test set:")

    print("RMSE:\t" + str(rmse))
    print("MAE:\t" + str(mae))
```

Obrázek 4.12: kód programové funkce `verification()`

Další funkcí, kterou jsem použil, je `plot_model()`, která na základě vstupních dat, vykreslí graf, který znázorňuje srovnání predikovaných dat oproti datům skutečným. Dále se vykreslí i histogram, který popisuje zastoupení vzorků v hladinách chybovosti.

```

#funkce pro vykreslení grafu a histogramu úspěšnosti predikce modelu
def plot_model(y,x,condition):

    f = plt.figure(figsize=(14,7))
    if condition:
        f.suptitle('Trénovací dataset')
    else:
        f.suptitle('Testovací dataset')
    ax1 = f.add_subplot(121)
    ax2 = f.add_subplot(122)

    ax1.scatter(y, x, edgecolors=(0, 0, 0))
    ax1.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=5)
    ax1.set(xlabel='Skutečná RUL', ylabel='Predikovaná RUL', title='RUL Skutečná vs. Predikovaná')

    x.columns = ['pred']
    y = pd.DataFrame(y)
    y.columns = ['RUL']
    histo = {
        "y_test": y.RUL,
        "y_pred": x.pred
    }
    histo = pd.DataFrame(histo)
    histo['error'] = abs(histo['y_test'] - histo['y_pred'])

    ax2.hist(histo['error'])
    ax2.set(xlabel='Absolutní chyba', ylabel='Počet predikovaných vzorků', title='Histogram absolutní chyby modelu')

```

Obrázek 4.13: kód programové funkce plot_model()

Poslední použitá funkce je funkce model(), která slouží pro snadné testování modelu. Vstupní proměnnou je již natrénovaný model. Na začátku této funkce je zavolána funkce load_data(), pro načtení trénovací a testovací datové sady do mezipaměti. Dále je model použit k předpovědi zbývajících životností na trénovací datové sadě a následně otestován a vizualizován do grafu a histogramu. Stejný postup je následně aplikován na testovací datové sadě.

```

#funkce pro trénování modelu
def model(model):
    load_data()
    train_pred = model.predict(x_train)
    train_pred = pd.DataFrame(train_pred)
    train_pred = train_pred.clip(lower=0)
    verification(y_train,train_pred,True)
    plot_model(y_train,train_pred,True)

    print()
    test_pred = model.predict(x_test)
    test_pred = pd.DataFrame(test_pred)
    test_pred = test_pred.clip(lower=0)
    verification(y_test,test_pred,False)
    plot_model(y_test,test_pred,False)

```

Obrázek 4.14: kód programové funkce model()

4.6.5 Trénování predikčního modelu

Pro trénování modelů se používají techniky strojového učení, což je podoblastí umělé inteligence, která se zabývá algoritmy a technikami které umožňují počítačovému systému 'učit se'. Strojové učení využívá algoritmy k identifikaci vzorů v datech a tyto vzory se pak používají k vytvoření datového modelu, který dokáže formulovat předpovědi. S větším množstvím dat a více zkušenostmi jsou výsledky strojového učení přesnější.

Srdcem algoritmu prediktivní údržby je detekční nebo predikční model. Tento model analyzuje extrahované indikátory stavu za účelem zjištění aktuálního stavu systému (detekce a diagnostika poruchy) nebo předpovědi jeho budoucího stavu (predikce zbývajících životností). [48]

Obecné rozdělení strojového učení:

- **Učení s učitelem** – Algoritmy strojového učení s učitelem se používají nejčastěji. U tohoto modelu plní člověk funkci průvodce, který učí algoritmus, jaké závěry má učinit. Při učení s učitelem je algoritmus trénován datovou sadou, která je již označena a má předdefinovaný výstup. Do této kategorie patří klasifikační a regresní metody strojového učení.
- **Učení bez učitele** – strojové učení bez učitele využívá nezávislejší přístup, ve kterém se počítač učí rozpoznat složité procesy a vzorce bez toho, aby mu člověk poskytoval bližší trvalé vedení. Strojové učení bez učitele zahrnuje školení založené na neoznačených datech nebo na konkrétním, definovaném výstupu. Do této kategorie patří shlukovací metoda strojového učení. [36]

Klasifikace algoritmů pro strojové učení:

- **Klasifikace** – vstupy jsou rozděleny do dvou nebo více tříd a vzdělávací systém musí vytvořit model schopný přiřadit jednu nebo více tříd mezi ty, které jsou k dispozici pro vstup.
- **Regrese** – pomáhají předpovídat nebo vysvětlovat konkrétní číselnou hodnotu na základě sady předchozích údajů.
- **Shlukování** – pomáhají předpovídat nebo vysvětlovat konkrétní číselnou hodnotu na základě sady předchozích údajů. [37]

Pro řešení této úlohy jsem je nejlepší možnost použít metodu učení s učitelem, protože je známý konkrétní výstup, který chceme, aby se model naučil. Pro předpověď zbývajících životností je lepší použít regresivní algoritmy, protože hledaná proměnná je reprezentována konkrétní číselnou hodnotou. Kdybychom chtěli předpovídat stav motoru reprezentovaný logickou jedničkou nebo nulou (normální/poruchový stav), bylo by vhodnější použít klasifikační algoritmy. Při trénování modelu jsem použil celkem šest různých modelů:

Lineární regrese

Cílem regrese je proložit naměřenými body přímkou, která nejlépe vystihuje vztah mezi proměnnými. Přímka prokládaná (většinou) pomocí metody nejmenších čtverců. Lineární regresní přímka má rovnici ve tvaru:

$$y = \beta_0 + \beta_1 * x_1 + \dots + \beta_n * x_n$$

kde:

- y = závislá proměnná
- x_n = nezávislé proměnné
- β_0 = Počátek přímky (intercept/počátek)
- β_n = koeficient sklonu přímky (gradient)

[38]

Dosažené výsledky:

```

Trénovací dataset:
RMSE: 19.7772973128328
MAE: 15.93331539321184

Testovací dataset:
RMSE: 37.085935328860316
MAE: 29.377253604360615

```

Obrázek 4.15: Výsledky lineární regrese

Polynomická regrese

Jedná se o model, který je lineární v parametrech, ale popisuje nelineární závislost mezi proměnnými. Tento model obsahuje pouze jednu nezávislou proměnnou, která se v něm však vyskytuje v různých mocninách a to od 1 do n , kde speciálním případem je polynom 1. stupně, který se rovná přímce. Tvar polynomu ovlivňuje také to, zda je stupeň polynomu lichý nebo sudý. Matematicky ho lze popsat rovnicí ve tvaru:

$$y = b_0 + b_1x + b_2x^2 + \dots + b_nx^n$$

kde:

- y = závislá proměnná
- x = nezávislá proměnná
- b_n = neznámé parametry, které chceme určit
- n = stupeň polynomu

[39]

Dosažené výsledky:

```

train set:
RMSE: 17.14459882183747
MAE: 13.18527945148721

test set:
RMSE: 35.28648555344051
MAE: 27.47119152852873

```

Obrázek 4.16: Výsledky polynomické regrese

SVR

Tato metoda vychází z metody podpůrných vektorů (SVM), ale v principu nejde o klasifikaci vstupních dat do určitých kategorií, ale o predikci reálných číselných hodnot. Metoda SVM je založena na principu rozdělení dat reprezentovaných vektory v mnoha-dimenzionálním příznakovém prostoru, do dvou skupin (tříd) s pomocí lineárního klasifikátoru – roviny tak, že data náležející odlišným třídám leží v opačných poloprostorech. Cílem je oddělit tyto odlišné třídy dat na co největší vzdálenost. [40]

Výhodou SVR je odolnost vůči odlehlým hodnotám v pozorování a poměrně rychlý výpočet. [40]

Dosažené výsledky:

```
Trénovací dataset:
RMSE: 17.01170008757814
MAE: 11.7802291750112

Testovací dataset:
RMSE: 37.835559080783
MAE: 28.871118133631377
```

Obrázek 4.17: Výsledky SVR

Rozhodovací strom

Rozhodovací strom je definován jako souvislý graf, který se skládá z kořene, větví, terminálních a neterminálních uzlů, kde kořen je vrchol grafu, který představuje otázku. Větev představuje odpověď na otázku položenou v kořeni a vedou do uzlů. Neterminální uzel představuje další otázku a vedou z něj další větve. Terminální uzel je konečný bod rozhodovacího stromu a je mu přiřazeno příslušné rozhodnutí. [41]

Dosažené výsledky:

```
Trénovací dataset:
RMSE: 0.0
MAE: 0.0

Testovací dataset:
RMSE: 41.19704909336616
MAE: 30.214722052535127
```

Obrázek 4.18: Výsledky rozhodovacího stromu

Náhodný les (random forest regressor)

Tento algoritmus je rozšířenou metodou rozhodovacího stromu, který kombinuje mnoho rozhodovacích stromů do jednoho modelu. V principu funguje tak, že vygeneruje několik rozdílných rozhodovacích stromů. Jednotlivé stromy běží paralelně nezávisle na sobě a výsledkem tohoto modelu je zprůměrovaný výstup jednotlivých stromů. Nevýhodou tohoto modelu je větší výpočetní náročnost. [42]

Dosažené výsledky:

```
Trénovací dataset:
RMSE: 10.285430029967266
MAE: 6.946560024706927

Testovací dataset:
RMSE: 37.54971647112861
MAE: 28.236737719313478
```

Obrázek 4.19: Výsledky náhodného lesa

Neuronová síť

Neuronové sítě napodobují strukturu lidské nervové soustavy. Základním prvkem je neuron. Neurony jsou navzájem propojeny a předávají si signály. Každý neuron může mít více vstupů ale jen jeden výstup. Každý vstup má určitou váhu a poté co neuron přijme vstupy, tak jejich hodnoty vynásobí váhami. Následně tyto součiny sečte, a pokud je výsledek větší než stanovený práh, tak se výsledek transformuje předem danou přenosovou funkcí a pošle na výstup.

[43]

Dosažené výsledky:

```
Trénovací dataset:  
RMSE: 17.01560740228125  
MAE: 12.542669307052819  
  
Testovací dataset:  
RMSE: 37.46470697285625  
MAE: 28.893948040182316
```

Obrázek 4.20: Výsledky neuronové sítě

5 Zhodnocení

5.1 Dosažené výsledky

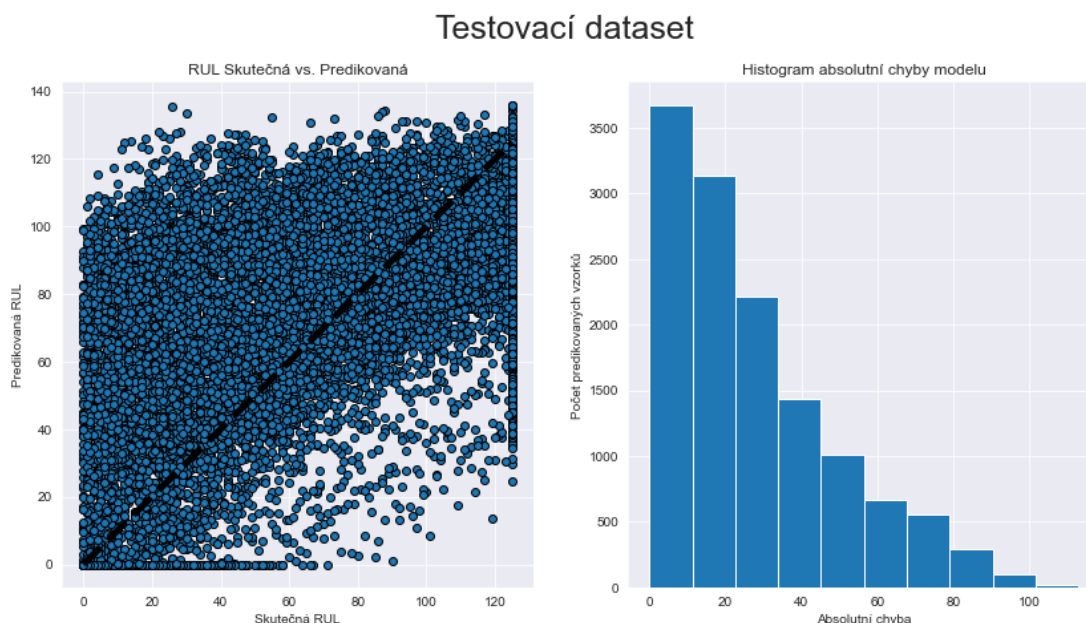
Výsledky jednotlivých modelů, lze nejlépe porovnat ve srovnávací tabulce:

Tabulka 5.1: srovnání jednotlivých modelů

model	RMSE	MAE
lineární regrese	37,056	29,377
polynomická regrese	35,286	27,471
SVR	37,836	28,871
rozhodovací strom	41,197	30,215
náhodný les	37,55	28,237
neuronová síť	37,465	28,894

Nejlepšího výsledku jsem dosáhl s polynomickou regresí 2. stupně, kde jsem na testovací datové sadě dosáhl RMSE = 35,286 a průměrné absolutní chyby = 27,471. Model predikoval hodnotu zbývajících životnosti motoru, která se pohybuje v rozmezí od 125 do 0 hodin.

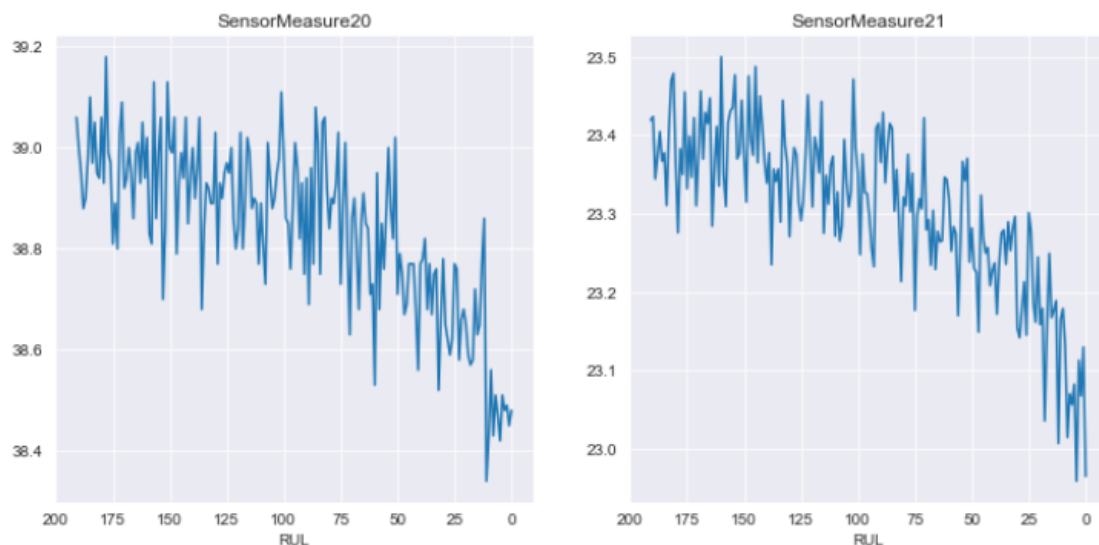
Na obrázku níže je vlevo graf, který zobrazuje predikované zbývajících životnosti v závislosti na skutečné hodnotě a vpravo je histogram, který zobrazuje počty predikovaných vzorků v závislosti na jejich absolutní chybě.



Obrázek 5.1: Výpis předpovězených hodnot oproti skutečným v grafech

Důvodem proč nejlepšího výsledku dosáhla právě polynomická regrese je podle mě ten, že výstupy ze senzorů, lze v průběhu měření popsat vzestupnou nebo sestupnou křivkou a polynomická

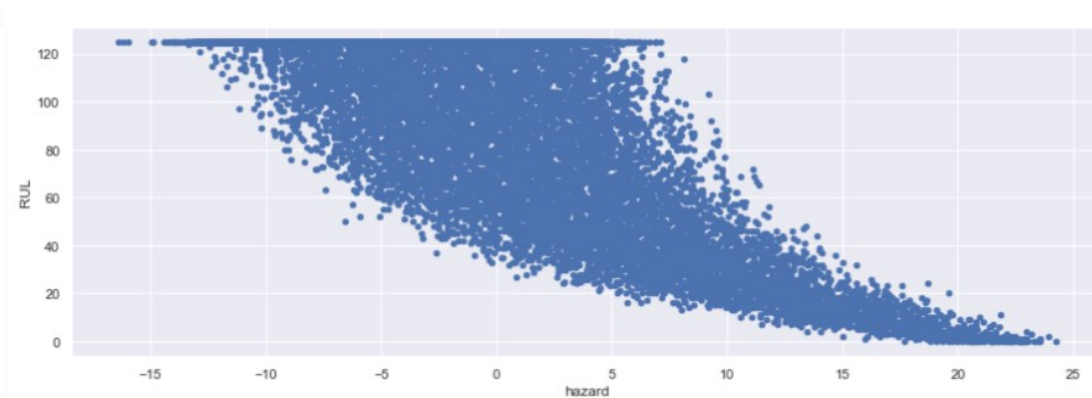
regrese 2. stupně se dokáže nejlépe této křivce přizpůsobit, a tedy nejsprávněji předvídat zbývající životnost na základě vstupních dat z těchto senzorů.



Obrázek 5.2: Ukázka dat ze senzorů vůči zbývající životnosti v grafech

5.2 Srovnání s referenčním řešením

Autor referenčního řešení [44] dosáhl nejlepšího výsledku $RMSE = 20,54$ pomocí modelu SVR. Důvodem proč autor dosáhl výrazně výsledků je dáno hlavně tím, že tento autor použil mnohem sofistikovanější metody předzpracování dat, jako je např. Coxův model proporcionálních rizik, ten vypočítává poměr nebezpečnosti, přičemž označuje například riziko selhání.



Obrázek 5.3: Výpis rizika selhání oproti zbývající životnosti v grafu [44]

Dále autor přidal do datové sady hodnoty "breakdown", která představuje, zda je motor v pořádku, nebo zda došlo ke zhroucení a hodnotu "start", která počítá dobu běhu motoru. Tyto hodnoty posléze využívá v trénování modelů.

s_20	s_21	RUL	breakdown	start
38.49	22.9735	4	0	195
38.30	23.1594	3	0	196
38.44	22.9333	2	0	197
38.29	23.0640	1	0	198
38.37	23.0522	0	1	199

Obrázek 5.4: Ukázka výpisu datové sady referenčního řešení [44]

Dalším důvodem, proč se mé výsledky výrazně liší je, že metrika RMSE je citlivá na odlehlé hodnoty, protože účinek každé chyby je úměrný velikosti druhé mocniny této chyby, tedy větší chyby mají nepřiměřeně velký účinek na RMSE.

Doporučil bych tedy v rámci předzpracování analýzu datového souboru identifikaci a vyloučení odlehlých hodnot.

5.3 Jak dosáhnout lepšího výsledku

Jak dosáhnout lepších výsledků je možné shrnout ve třech krocích:

Předzpracování dat - Největší vliv na celkový výsledek predikce má tato fáze zpracování dat. Použití pokročilých technik jako je např. Coxův model proporcionálních rizik umožní mnohem lépe předvídat hodnotu zbývajících životnosti a díky tomu přesněji reflektovat skutečný stav motoru. Použití této metody však přesahovalo rámec mé bakalářské práce.

Parametrizace modelu jednotlivé algoritmy mají širokou škálu parametrů, jež ovlivňují výsledný model. Doporučeným řešením je tedy napsat algoritmus, který otestuje co nejvíce možností a vybere z nich tu nejlepší. Rovněž aplikace této metody přesahovala rámec mé bakalářské práce.

Objem trénovacích dat - obecně v oblasti strojového učení platí, že čím více trénovacích dat modelu poskytneme, tím lepších výsledků dosahuje. V tomto případě to ovšem není možné, protože máme předem stanovená data jak pro trénování, tak pro testování modelů.

Závěr

Cílem této práce bylo seznámit se v teoretické části s moderními nástroji pro zpracování dat, jako je např. strojové učení a v praktické části využití těchto nástrojů k analýze dat z výrobního procesu v aplikační oblasti prediktivní údržby.

Teoretická část je rozdělena do tří kapitol. V první kapitole jsem provedl popis a rozbor konceptu Průmysl 4.0, včetně seznámení se s klíčovými technologiemi, které jsou s tímto konceptem spojovány jako jsou např. "internet věcí" nebo "digitální dvojče". Ve druhé kapitole jsem se zaměřil na data produkovaná ve výrobním procesu, a to jak na způsoby sběru dat, tak na typy dat, které se vyskytují v průmyslu a také technologie, jež figurují ve schématu měřicího řetězce, jako jsou např. snímače nebo měřicí přístroje. Poslední kapitola teoretické části je zaměřená na nástroje, které se využívají ke zpracování dat. V této kapitole jsem shrnul historický vývoj těchto nástrojů a blíže popsal moderní nástroje, jako jsou např. SW nástroj firmy National Instruments DIAdem nebo vývojové prostředí Jupyter Notebook využívající programovací jazyk Python.

Praktická část je rozdělena do dvou kapitol. První kapitola je věnována praktickému použití nástrojů popsaných v teoretické části, kde aplikace těchto nástrojů se skládá z několika kroků. Mezi tyto kroky patří nalezení a výběr vhodné datové sady pro použití v oblasti prediktivní údržby, kde jsem vybral datovou sadu simulovaných leteckých motorů, jež obsahuje záznamy běhu motoru, až do bodu jeho selhání. Dalším krokem je popis struktury vybrané datové sady spolu se stanovením cílů analýzy, kde cílem analýzy bylo určit kritické parametry a následně z nich vytvořit model pro monitoring stavu motoru a predikci poruchových stavů.

Dále následoval výběr nástroje pro analýzu datové sady, kde jsem se rozhodl pro použití vývojového prostředí Jupyter Notebook a programovacího jazyka Python a prostřednictvím jeho knihoven funkcí pro strojové učení jsem následně vytvořil model, který předpovídá zbývající životnost motorů.

Posledním krokem je provedení této analýzy ve vybraném nástroji, jež se skládá z předzpracování dat, identifikace indikátorů stavu a trénování predikčního modelu za použití metod strojového učení. Ve druhé kapitole praktické části jsem shrnul a vizualizoval dosažené výsledky. Dále jsem porovnal mé postupy s referenčním řešením, abych validoval své hypotézy a výsledky. Dále jsem vysvětlil rozdíly v porovnání mých výsledků s výsledky referenčního řešení. Nakonec jsem shrnul případný postup, jak by šla má práce vylepšit za účelem dosažení lepších výsledků přibližujících se výsledkům referenčního řešení, jako je například použití pokročilých technik ve fázi předzpracování dat nebo využití algoritmu, který otestuje širokou škálu parametrů modelu při jeho trénování.

Všechny body zadání byly splněny. Prokázal jsem možnost praktického použití nástrojů popsaných v teoretické části. V praktické části jsem průběžně validoval své hypotézy a výsledky s úvahami výsledky referenčního řešení a dokázal jsem vysvětlit rozdíly mých výsledků s výsledky referenčního řešení. Dále jsem prokázal potenciál moderních nástrojů pro zpracování dat i pro analýzu dat sbíraných v rámci moderních digitalizovaných výrobních procesů a ověřil možnost využití analýzy dat v aplikační oblasti prediktivní údržby.

Použitá literatura

- [1] Vývoj průmyslových revolucí. Hackernoon.com [online]. [cit. 2021-4-29]. Dostupné z: <https://hackernoon.com/industry-40s-ultimate-impact-on-manufacturing-business-rq9f3y7k>
- [2] Anglická průmyslová revoluce a její rozšíření. Www.dejepis.com [online]. [cit. 2020-09-28]. Dostupné z: https://www.dejepis.com/ucebnice/anglicka-prumyslova-revoluce-a-jeji-rozsireni/#Predpoklady_pro_prumyslovou_revoluci
- [3] Od 1. průmyslové revoluce ke 4. Www.technickytydenik.cz [online]. [cit. 2020-09-28]. Dostupné z: https://www.technickytydenik.cz/rubriky/ekonomika-byznys/od-1-prumyslove-revoluce-ke-4_31001.html
- [4] Průmyslová revoluce – Od Průmyslu 1.0 k Průmyslu 4.0. Www.desouttertools.cz [online]. [cit. 2020-09-28]. Dostupné z: <https://www.desouttertools.cz/prumysl-4-0/novinky/563/prumyslova-revoluce-od-prumyslu-1-0-k-prumyslu-4-0>
- [5] YÁÑEZ, Fran. The 20 Key Technologies of Industry 4. 0 and Smart Factories: The Road to the Digital Factory of the Future. 1. Galicie Španělsko: Independently published, 2017. ISBN 978-1973402107.
- [6] Vývoj průmyslových revolucí. Microsoft [online]. [cit. 2021-4-29]. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-is-virtualization/>
- [7] DEAN, Jared. Big data, data mining and machine learning: value creation for business leaders and practitioners. Hoboken, New Jersey: John Wiley and Sons, [2014]. ISBN 978-1-118-92069-5.
- [8] Co jsou big data. Www.napocitaci.cz [online]. [cit. 2020-11-15]. Dostupné z: <https://www.napocitaci.cz/33/co-jsou-big-data-uniqueidgOkE4NvrWuNY54vrLeM67z9XeP5DbI7sT6QQQea68B8/>
- [9] History of Big Data. Www.sas.com [online]. [cit. 2020-11-15]. Dostupné z: https://www.sas.com/cs_cz/insights/big-data/what-is-big-data.html
- [10] IO-Link – popis digitální komunikace pro senzory. Www.automatizace.hw.cz [online]. [cit. 2020-11-15]. Dostupné z: <https://automatizace.hw.cz/iolink-popis-digitalni-komunikace-pro-senzory>
- [11] Bezdrátové zemské systémy 5. generace (5G) [online]. březen 2017 [cit. 2020-11-17]. Dostupné z: <https://www.ctu.cz/sites/default/files/obsah/stranky/147034/soubory/170330bezdratovezemskesystemy5.generace5gv.2.pdf>
- [12] SCADA. Promotic [online]. [cit. 2021-4-29]. Dostupné z: <https://www.promotic.eu/cz/pmdoc/WhatIsPromotic/WhatIsScada.htm>

- [13] SCADA. Tulip.co [online]. [cit. 2021-4-29]. Dostupné z: <https://tulip.co/blog/digital-transformation/what-is-a-smart-factory-and-what-it-means-for-you/>
- [14] Základy měření elektrických a neelektrických veličin v budovách a bytech. Http://fei1.vsb.cz/ [online]. [cit. 2020-12-18]. Dostupné z: http://fei1.vsb.cz/kat420/vyuka/Bakalarske/STB/3_mereni_neelektrickych_velicin.pdf
- [15] Dubec, M., Skotnica, J., Senzory a měření. Elektronické skriptum, VŠB Ostrava, 2012, 318 stran
- [16] Wittassek, T., Virtuální instrumentace I.. Elektronické skriptum, VŠB Ostrava, 2012, 264 stran
- [17] Zpracování dat. Tulip.co [online]. [cit. 2021-4-29]. Dostupné z: <https://managementmania.com/cs/zpracovani-dat-data-processing>
- [18] Zpracování dat. Wikisofia.cz [online]. [cit. 2021-4-29]. Dostupné z: https://wikisofia.cz/wiki/Charakterizujte_tabulkov%C3%BD_procesor_a_jeho_z%C3%A1kladn%C3%AD_risy_a_funkce._Uve%C4%8Fte_typick%C3%A9_situace,_kdy_je_vhodn%C3%A9_vyu%C5%BE%C3%ADt_tabulkov%C3%BD_procesor._Popi%C5%A1te_pou%C5%BEit%C3%AD_vzorc%C5%AF_v_tabulkov%C3%A9m_procesoru.
- [19] Tabulkové procesory. Wikisofia.cz [online]. [cit. 2021-4-29]. Dostupné z: <https://office.lasakovi.com/excel/>
- [20] Tabulkové procesory. Muni.cz [online]. [cit. 2021-4-29]. Dostupné z: <http://boss.ped.muni.cz/Datab%C3%A1ze1.pdf>
- [21] Doc. Ing. Jan Židek, CSc. SW NÁSTROJE MANAGEMENTU NAMĚŘENÝCH DAT [online]. [cit. 2021-3-05]. Učební text. Katedra elektrických měření, Fakulta elektrotechniky a informatiky, VŠB-TU Ostrava.
- [22] Prediktivní údržba v digitalizovaném průmyslu. Http://www.act-in.cz/ [online]. [cit. 2020-12-18]. Dostupné z: <http://www.act-in.cz/nase-publikace/prediktivni-udrzba-v-digitalizovanem-prumyslu>
- [23] Prediktivní údržba, cesta ke snížení nákladů. Https://www.technickytydenik.cz [online]. [cit. 2020-12-18]. Dostupné z: https://www.technickytydenik.cz/rubriky/archiv/prediktivni-udrzba-cesta-ke-snizeni-nakladu_18834.html
- [24] BIG DATA IN AVIATION - REDUCE COSTS THRU PREDICTIVE MAINTENANCE. Https://www.exsyn.com [online]. [cit. 2020-12-18]. Dostupné z: <https://www.exsyn.com/blog/big-data-in-aviation-predictive-maintenance>
- [25] Prediktivní údržba: Analýza dat senzorů výrobních linek a strojů pro redukci odstávek a zvýšení kvality produktů. Http://ce.almanachprodukce.cz [online]. [cit. 2020-12-18]. Dostupné z:

<http://ce.almanachprodukce.cz/media/down/materialy/pdf/a678b6e0d2c717a280f3ff771db719f6.pdf>

- [26] Databázové systémy. Reddit.com [online]. [cit. 2021-4-29]. Dostupné z: <https://www.reddit.com/r/datasets/>
- [27] NASA data repository. Nasa.gov [online]. [cit. 2021-4-29]. Dostupné z: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>
- [28] B. Saha and K. Goebel (2007). "Battery Data Set", NASA Ames Prognostics Data Repository (<http://ti.arc.nasa.gov/project/prognostic-data-repository>), NASA Ames Research Center, Moffett Field, CA
- [29] J. Lee, H. Qiu, G. Yu, J. Lin, and Rexnord Technical Services (2007). IMS, University of Cincinnati. "Bearing Data Set", NASA Ames Prognostics Data Repository (<http://ti.arc.nasa.gov/project/prognostic-data-repository>), NASA Ames Research Center, Moffett Field, CA
- [30] B. Bole, C. Kulkarni, and M. Daigle, 'Adaptation of an Electrochemistry-based Li-Ion Battery Model to Account for Deterioration Observed Under Randomized Use', Annual Conference of the Prognostics and Health Management Society, 2014
- [31] Biao Wang, Yaguo Lei, Naipeng Li, Ningbo Li, "A Hybrid Prognostics Approach for Estimating Remaining Useful Life of Rolling Element Bearings", *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 401-412, 2020. DOI: 10.1109/TR.2018.2882682.
- [32] A. Saxena and K. Goebel (2008). "Turbofan Engine Degradation Simulation Data Set", NASA Ames Prognostics Data Repository (<http://ti.arc.nasa.gov/project/prognostic-data-repository>), NASA Ames Research Center, Moffett Field, CA
- [33] M. Chao, C. Kulkarni, K. Goebel and O. Fink (2021). "Aircraft Engine Run-to-Failure Dataset under real flight conditions", NASA Ames Prognostics Data Repository (<http://ti.arc.nasa.gov/project/prognostic-data-repository>), NASA Ames Research Center, Moffett Field, CA
- [34] RMSE. Nasa.gov [online]. [cit. 2021-4-29]. Dostupné z: https://cs.qaz.wiki/wiki/Root-mean-square_deviation
- [35] RMSE. Qaz.wiki [online]. [cit. 2021-4-29]. Dostupné z: https://cs.qaz.wiki/wiki/Mean_absolute_error
- [36] Strokové učení. Qaz.wiki [online]. [cit. 2021-4-29]. Dostupné z: <https://www.oracle.com/cz/data-science/machine-learning/what-is-machine-learning/>
- [37] Machine learning. Oracle.com [online]. [cit. 2021-4-29]. Dostupné z: <https://bloginnovazione.it/cs/machine-learning/3716/>

- [38] Lineární regrese. Bloginnovazione.it [online]. [cit. 2021-4-29]. Dostupné z: <https://ksoc.ff.cuni.cz/wp-content/uploads/sites/76/2018/09/6.-Statistika2-Line%C3%A1rn%C3%AD-regrese.pdf>
- [39] Polynomická regrese. Cuni.cz [online]. [cit. 2021-4-29]. Dostupné z: <https://ach.upol.cz/download/133>
- [40] SVR. Upol.cz [online]. [cit. 2021-4-29]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=190575
- [41] Rozhodovací strom. Vutbr.cz [online]. [cit. 2021-4-29]. Dostupné z: https://wikisofia.cz/wiki/Rozhodovac%C3%AD_tabulky_a_stromy
- [42] Náhodný les. Wikisofia.cz [online]. [cit. 2021-4-29]. Dostupné z: <https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f>
- [43] Neuronové sítě. Medium.com [online]. [cit. 2021-4-29]. Dostupné z: <https://www.napocitaci.cz/33/neuronove-site-a-princip-jejich-fungovani-uniqueidgOkE4NvrWuNY54vrLeM670eFNQh552VdDDuIX7UDBY/>
- [44] Referenční řešení. Github.com [online]. [cit. 2021-4-29]. Dostupné z: <https://github.com/kpeters/exploring-nasas-turbofan-dataset>
- [45] Google dataset search. Google.com [online]. [cit. 2021-4-29]. Dostupné z: <https://datasetsearch.research.google.com/>
- [46] Kaggle. Google.com [online]. [cit. 2021-4-29]. Dostupné z: <https://www.kaggle.com/datasets>
- [47] Kaggle. Data.gov [online]. [cit. 2021-4-29]. Dostupné z: <https://catalog.data.gov/dataset>
- [48] Trénování predikčního modelu. Mathworks.com [online]. [cit. 2021-4-29]. Dostupné z: <https://www.mathworks.com/help/predmaint/gs/designing-algorithms-for-condition-monitoring-and-predictive-maintenance.html>

Seznam příloh

Příloha A: program vlastního řešení

- BP_PAL0232.ipynb

Příloha B: program referenčního řešení

- 1_FD001_EDA+baseline.ipynb
- 2_FD001_updated_RUL_assumption_and_SVR.ipynb
- 3_FD001_time_series.ipynb
- 4_FD001_Survival_analysis.ipynb

Příloha C: datová sada

- RUL_FD001.txt
- test_FD001.txt
- train_FD001.txt